Integrative analysis of data from multiple experiments

DISSERTATION
zur Erlangung des akademischen Grades
Doctor of Philosophy
(Ph. D.)

eingereicht an der Lebenswissenschaftlichen Fakultät der Humboldt-Universität zu Berlin

von
Sivilingeniør Jonathan Ronen,

Präsidentin der Humboldt-Universität zu Berlin
Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Lebenswissenschaftlichen Fakultät der Humboldt-Universität zu Berlin
Prof. Dr. Bernhard Grimm

Gutachter/innen:

1. **Dr. Altuna Akalin**

2. **Prof. Dr. Nikolaus Rajewsky**

3. **Prof. Dr. Ulf Leser**

Tag der mündlichen Prüfung: 13.07.2020

Abstract

The development of high throughput sequencing (HTS) was followed by a swarm of protocols utilizing HTS to measure different molecular aspects such as gene expression (transcriptome), DNA methylation (methylome) and more. This opened opportunities for developments of data analysis algorithms and procedures that consider data produced by different experiments.

Considering data from seemingly unrelated experiments is particularly beneficial for Single cell RNA sequencing (scRNA-seq). scRNA-seq produces particularly noisy data, due to loss of nucleic acids when handling the small amounts in single cells, and various technical biases. To address these challenges, I developed a method called *netSmooth*, which de-noises and imputes scRNA-seq data by applying network diffusion over a gene network which encodes expectations of co-expression patterns. The gene network is constructed from other experimental data. Using a gene network constructed from protein-protein interactions, I show that *netSmooth* outperforms other state-of-the-art scRNA-seq imputation methods at the identification of blood cell types in hematopoiesis, as well as elucidation of time series data in an embryonic development dataset, and identification of tumor of origin for scRNA-seq of glioblastomas. *netSmooth* has a free parameter, the diffusion distance, which I show can be selected using data-driven metrics. Thus, *netSmooth* may be used even in cases when the diffusion distance cannot be optimized explicitly using ground-truth labels.

Another task which requires in-tandem analysis of data from different experiments arises when different omics protocols are applied to the same biological samples. Analyzing such *multi-omics* data in an integrated fashion, rather than each data type (RNA-seq, DNA-seq, etc.) on its own, is benefitial, as each omics experiment only elucidates part of an integrated cellular system. The simultaneous analysis may reveal a comprehensive view. I developed a method called *maui*, to find latent factor representations of multi-omics data. The method uses a variational autoencoder to learn nonlinear patterns in different omics data types, and produces latent factor representations which capture meaningful biology. I demonstrate its applicability on multi-omics data from colorectal cancer (CRC) tumors and cancer cell lines. Latent factor representations produced by *maui* are predictive of patient survival, and they allow patients to be clustered into molecular sub-types in a way which partly recreates the current gold-standard for CRC sub-typing; moreover, I show that one of the current gold standard sub-types

might need to be split into two groups of patients, with distinct survival probabilities and dysregulation of different molecular pathways. I also used *maui* to quality-control colorectal cancer cell lines; by quantifying the similarity of cancer cell lines to primary tumors, i made predictions as to which cell lines are more appropriate models for the different CRC sub-types.

Finally, application of *netSmooth* prior to feeding data to *maui* for multi-omics integration further improves the survival prediction capabilities of the method.

netSmooth is an R package and is obtainable from Bioconductor. *maui* is a python package, and is available from PyPI.

Zusammenfassung

Auf die Entwicklung der Hochdurchsatz-Sequenzierung (HTS) folgte eine Reihe von speziellen Erweiterungen, die erlauben verschiedener zellbiologischer Aspekte wie Genexpression, DNA-Methylierung, etc. zu messen. Die Analyse dieser Daten erfordert die Entwicklung von Algorithmen, die einzelne Experimenteberücksichtigen oder mehrere Datenquellen gleichzeitig in betracht nehmen.

Der letztere Ansatz bietet besondere Vorteile bei Analyse von einzelligen RNA Sequenzierung (scRNA-seq) Experimenten welche von besonders hohem technischen Rauschen, etwa durch den Verlust an Molekülen durch die Behandlung geringer Ausgangsmengen, gekennzeichnet sind. Um diese experimentellen Defizite auszugleichen, habe ich eine Methode namens *netSmooth* entwickelt, welche die scRNA-seq-Daten entrascht und fehlende Werte mittels Netzwerkdiffusion über ein Gennetzwerk imputiert. Das Gennetzwerk reflektiert dabei erwartete Koexpressionsmuster von Genen. Unter Verwendung eines Gennetzwerks, das aus Protein-Protein-Interaktionen aufgebaut ist, zeige ich, dass *netSmooth* anderen hochmodernen scRNA-Seq-Imputationsmethoden bei der Identifizierung von Blutzelltypen in der Hämatopoese, zur Aufklärung von Zeitreihendaten unter Verwendung eines embryonalen Entwicklungsdatensatzes und für die Identifizierung von Tumoren der Herkunft für scRNA-Seq von Glioblastomen überlegen ist. *netSmooth* hat einen freien Parameter, die Diffusionsdistanz, welche durch datengesteuerte Metriken optimiert werden kann. So kann *netSmooth* auch dann eingesetzt werden, wenn der optimale Diffusionsabstand nicht explizit mit Hilfe von externen Referenzdaten optimiert werden kann.

Eine integrierte Analyse ist auch relevant wenn *multi-omics* Daten von mehrerer Omics-Protokolle auf den gleichen biologischen Proben erhoben wurden. Hierbei erklärt jeder einzelne dieser Datensätze nur einen Teil des zellulären Systems, während die gemeinsame Analyse ein vollständigeres Bild ergibt. Ich entwickelte eine Methode namens *maui*, um eine latente Faktordarstellungen von *multi-omics* Daten zu finden. Das Verfahren verwendet einen Variational Autoencoder, der nichtlineare Muster in verschiedenen Omics-Datentypen zu lernen die biologisch interpretierbar sind.

Ich demonstriere seine Anwendbarkeit auf *multi-omics* Daten von Darmkrebs-(CRC)-Tumoren und Krebszelllinien. Die von *maui* produzierten latenten Faktor sind prädiktiv für das Patientenüberleben, und ermöglichen es, Patienten in molekulare Subtypen

zu gruppieren, so dass sie teilweise den aktuellen Goldstandard für die CRC Subtypisierung nachbilden. Ein CRC-Subtyp ließ sich in zwei Gruppen von Patienten aufteilen, welche durch unterschiedliche Überlebenswahrscheinlichkeiten und Dysregulation verschiedener molekularer Pfade gekennzeichnet sind sind. Ich habe auch *maui* zur Qualitätskontrolle von Darmkrebs-Zelllinien verwendet, um festzustellen, welche Zelllinien auf Grund ihrer Ähnlichkeit mit CRC-Subtypen am besten als Modelle für Drug Discovery Studien geeignet sind.

Die Vorverarbeitung der Daten durch *netSmooth* und anschließende Verwendung von *maui* verbessert die Vorhersage der Überlebenswahrscheinlichkeit. *netSmooth* ist ein R-Paket und kann bei Bioconductor bezogen werden. *maui* ist ein Python-Paket und ist über PyPI erhältlich.

Author Contributions

Parts of this dissertation have been released before in the following publications:

1. Ronen, Jonathan, and Altuna Akalin. "netSmooth: Network-smoothing based imputation for single cell RNA-seq." F1000Research 7 (2018).
2. Ronen, Jonathan, Sikander Hayat, and Altuna Akalin. "Evaluation of colorectal cancer subtypes and cell lines using deep learning." bioRxiv (2018): 464743.
3. Altuna Akalin, Vedran Franke, Bora Uyar, and Jonathan Ronen. "Computational Genomics with R." Berlin 2019 (in press)

Section 1.4 is adapted from Akalin, Franke, Uyar, and Ronen (2019), with large sections reproduced verbatim. Chapter 2 is reproduced with minor edits from Ronen and Akalin (2018a). Chapter 3 is reproduced with minor edits from Ronen, Hayat, and Akalin (2018). Figure captions state when figures are reproduced from any of these publications.

Author contributions for the publications are as follows:

(Ronen and Akalin, 2018a) AA conceptualized the project, AA and JR conceived of the algorithm together. All the analysis and software development was done by JR, who also wrote the initial draft of manuscript with input from AA. AA supervised the writing, software development and analysis. JR wrote R package with input and code review and contributions from AA.

(Ronen et al., 2018) AA, SH and JR conceptualized the project. AA set the objectives with input from JR and SH. The analysis presented in this manuscript, and all the software development was done by JR. SH provided additional analysis on latent factor interpretation and cell line matching. AA and JR wrote initial draft of the manuscript with input from SH. AA supervised the writing, software development and analysis. JR, AA and SH reviewed and finalized the manuscript.

(Akalin et al., 2019) JR contributed the chapter on multi-omics data integration, with input from AA. AA edited the text.

Erklärung

Hiermit erkläre ich, die Dissertation selbstständig und nur unter Verwendung der angegebenen Hilfen und Hilfsmittel angefertigt zu haben.

Ich habe mich anderwärts nicht um einen Doktorgrad beworben und besitze keinen entsprechenden Doktorgrad.

Ich erkläre, dass ich die Dissertation oder Teile davon nicht bereits bei einer anderen wissenschaftlichen Einrichtung eingereicht habe und dass sie dort weder angenommen noch abgelehnt wurde.

Ich erkläre die Kenntnisnahme der dem Verfahren zugrunde liegenden Promotionsordnung der Lebenswissenschaftlichen Fakultät der Humboldt-Universität zu Berlin vom 5. März 2015.

Weiterhin erkläre ich, dass keine Zusammenarbeit mit gewerblichen Promotionsbearbeiterinnen/Promotionsberatern stattgefunden hat und dass die Grundsätze der Humboldt-Universität zu Berlin zur Sicherung guter wissenschaftlicher Praxis eingehalten wurden.

Berlin, _____ _____

Jonathan Ronen

Acknowledgments

I WOULD LIKE to take this opportunity to thank my parents Jill and David, without whose support I would never have come half as far, and my sister Britt, who's been with me for most of my life. I am especially grateful of my father's personal example, attending college at 50, a reminder that it's never to late to grow. I would like to thank my advisor, Altuna, for invaluable guidance in shaping this work; for finding challenging projects for me and enabling me to grow. I would like to thank my colleagues Vedran and Bora, for all they have taught me and the drinks they shared with me. I would like to thank my committee members, Claudia and Nikolaus, for guidance, support, and advice on this work.

The results shown here are in part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga.

And most of all, I thank my wife Ella, without whom I don't even know...

For poi.

Contents

1

Introduction

Since the discovery of the structure of DNA and the birth of the field of molecular biology that followed, new discoveries and experimental techniques have made biology research astonishingly high tech. Especially since the completion of the Human Genome Project nearly two decades ago, the proliferation of high throughput sequencing and a plethora of experimental protocols have lead to a renaissance in the life sciences. The nature of the molecular systems under study and the technologies employed have resulted in

troves of experimental data which, while undeniably informative, is also rife with bias and high variance. These developments have come in tandem with the development of computational biology, as statistical methods have been neccesitated by both the volume and other challenges with the data.

A common theme throughout the developments of computational methods has been to share information across different observations to tease out the patterns in spite of noise and technical bias. Soon after practical DNA sequencing was invented (Sanger et al., 1977; Maxam and Gilbert, 1977) in ground-breaking work that won Frederick Sanger his second nobel prize in chemistry*, computational tools were taken advantage of. Sanger sequencing is only effective for fragments of limited length, but biologists soon set their sights on sequencing whole genomes. By sequencing many small, randomly sheared, overlapping DNA fragments, in a process termed *shotgun sequencing*, longer sequences may be assembled by aligning fragment sequences based on their overlapping segments. The potential of algorithms was realized quickly therafter (Staden, 1979), when they were used for de-novo sequence assembly of bacteriophage genomes. Together with Sanger sequencing, shotgun sequencing genome assembly algorithms (Batzoglou et al., 2002) formed the basis for the Human Genome Project (HGP). The reference human genome eventually produced by the HGP (International Human Genome Sequencing Consortium et al., 2004) represents a consensus sequence of DNA donated by many volunteers from a diverse population†. This makes the human reference genome itself the result of combining data from many different sequencing experiments. These developments set the stage for the now flourishing field of computational biology, where philosophically similar problems and solutions are common: sharing information across experiments, considering uncertainty and, where possible,

*His first was for his work on protein sequence.

†In order to protect their privacy, many more donors were recruited whose samples were never sequenced.

correcting errors.

As Sanger sequencing was replaced by Next Generation Sequencing (NGS) (Ronaghi et al., 1996; Margulies et al., 2005), a wealth of sequencing protocols have been developed, allowing researchers to probe molecular aspects other than the DNA sequence, e.g. gene expression or DNA methylation, using the same technology. Each such protocol came with corresponding developments in bioinformatics, and often following the theme where an algorithm attempts to *reconstruct the forest from the trees*, i.e. infer the bigger picture by combining many noisy observations into a coherent story.

Prominent examples of this class of algorithm are frequently used in analyzing RNA sequencing, ChIP sequencing, and Bisulfite sequencing (all described in more detail in Section 1.2 on page 7). In RNA sequencing (RNA-seq) experiments, the goal is often to determine differential gene expression under different conditions, e.g. a mutant vs. the wild type, or a treated group vs. a control group (Oshlack et al., 2010). The noisy nature of both biological systems and RNA-seq experiments means there will inevitably be some variance in expression patterns between equally treated samples. In order to quantify differential expression in the presense of such noise, statistical methods are employed, which learn patterns across technical and biological replicates of the same conditions. By sharing parameters in statistical distributions among e.g. different genes or different samples, the variance of the different estimates can be reduced, and biological and technical variability can better be quantified. Chromatin Immunoprecipiation (ChIP) sequencing can be used to profile epigenetic states such as histone modifications (Barski et al., 2007). When different chromatin marks are examined together, they may be used to define genome-wide *chromatin states* (Mikkelsen et al., 2007), i.e. *active* or *silent* chromatin domains, which can be inferred from the makeup of different histone modifications. The different histone modifications can be read in separate ChIPs, and algorithms (Ernst and Kellis, 2012) are used to statisti-

cally infer the chromatin state of the element of interest (promoters, enhancers, etc). In fact, most functional genomic elements, such as genes, promoters, and enhancers themselves were discovered by integrating data from many experiments and predicting the presence of such elements algorithmically, such as was done throughout the ENCODE project (ENCODE Project Consortium et al., 2004, 2007). This is a prime example of combining data from different sequencing experiments of the same (or similar) samples, leading to insights which would not be possible otherwise. Other epigenetic assays such as Bisulfite sequencing or ATAC sequencing (also described in more detail in Section 1.2 on the next page) may also be used to enrich the picture of the epigenetic landscape, and improve inferences, when data from a multitude of experiments is combined.

In my doctoral studies, I have continued this very tradition. In the pages that follow, I will describe two computational methods I have developed as part of my doctoral work. The first is a method to reduce bias and variance in whole genome assays, using informative priors about genes' interaction patterns, which are learned from thousands of previous experiments. The second method is a way to integrate multi-omics data, utilizing nonlinear patterns within and across different data modalities and finding succinct representations of the data which empower downstream analysis. I show the applicability of the methods in analyzing single cell RNA sequencing, cancer subtyping, and quality control of cancer models such as cell lines. I also show that using the two methods in tandem, integrating data from both previous experiments and multi-omics experiments, results in improved clinical relevance of cancer subtypes. Taken together, this dissertation makes up a body of work which makes a modest contribution to the field.

1.1 Thesis outline

In the following sections, I will introduce common genomic assays and data types generated by those assays. Then I will introduce in more detail two problem areas where I have made contributions. Section 1.3 on page 12 will introduce Single Cell RNA sequencing (scRNA-seq) and the drop-out problem, and in section 1.4 on page 16 I will present a review of computational methods for multi-omics data integration, with a focus on matrix factorization methods. The content is largely reproduced from (Akalin, Franke, Uyar, and Ronen, 2019), where I authored the chapter on multi-omics data integration. It has been edited for the overall clarity of this dissertation. Chapter 2 describes a novel method for dealing with drop-outs in scRNA-seq by integrating data from multiple experiments. The method was published in (Ronen and Akalin, 2018a), from which the content is reproduced, with some editing for clarity in the context of this dissertation. Then, in chapter 3, I present another method I developed, a deep learning-based latent vaiable model for multi-omics data integration, with applications to cancer sub-typing. The content is reproduced from (Ronen, Hayat, and Akalin, 2018). Finally, I discuss the overall impact of the work and share some concluding remarks in chapter 4.

1.2 Genomic assays and data types

The postfix *ome* in molecular biology, such as in gen*ome*, implies a completeness of some class. For instance, a transcriptome covers all the RNA *transcripts* in a biological sample, and the proteome describes the totality of proteins. Hence, omics refers to comprehensive, or total assays, where an entire class of something is characterized; e.g. transcriptomics refers

to the characterization of the RNA transcripts in a sample, metabolomics to the profiling of metabolites, etc.

In this section, I will briefly introduce some of the most common genomic assays and the data types they produce.

DNA sequencing is a way to determine the sequence of nucleotides in DNA. First invented in the 1970s and having inspired the world in the 1990s and early 2000s through the Human Genome Project (HGP), today DNA sequencing is an indispensible part of modern research in biology. Of the so-called second generation sequencing, which was developed in the 1990s as a result of the HGP, the most commonly used today is *sequencing by synthesis*. First, long DNA molecules are cut into shorter fragments using transposase. Then, adapters are ligated at the cut sites, which enable the fragments to hybridize with oligonucleotides in the sequencing flow cell, holding them in place. The fragments are then denatured, so that the forward and reverse strands separate, and polymerases are used to make many copies of each strand (amplification). Finally, primers are added to the DNA fragments which enable polymerases to add special fluorescent tagged nucleotides with a terminator which enables the process to be controlled and only happen one nucleotide at a time. Each of the four possible bases is given a unique fluorescent color, and the sequence can be read by observing the color at each cycle. This is done for many different molecules in parallel, enabling high throughput sequencing of an entire genome in hours (whole genome sequencing, WGS). Sometimes, as a cost saving measure, only the coding sequence of the genome is sequenced; this is referred to as whole exome sequencing (WES). Once the genomic sequence of a sample is determined, we can compare it to a reference sequence, and thus determine the presence or absence of single nucleotide

polymorphisms (SNPs), insertions, deletions, and larger structural variations such as copy number variations (CNV) or translocations. When used in this way to assert the presence of mutations in a disease sample, both WGS and WES data may be represented as a list of events (SNPs, insertions, deletions) and their coordinates, or, as is done later in this text for analysis purposes, the mutation data may be represented as an $N_{\text{genes}} \times N_{\text{samples}}$ binary matrix, where $m_{ij} = 1$ iff. sample j has a deleterious mutation in gene i, and $m_{ij} = 0$ otherwise. CNVs may be determined in e.g. a tumor by comparing the read depth of a tumor sample to that of a normal (non-tumor) sample. In this way, deletions and amplifications may be discovered. The genome may then be divided into segments which are deleted or amplified. Later in the text, I represent such data as an $N_{\text{segments}} \times N_{\text{samples}}$ matrix C where c_{ij} is the copy number of segment i in sample j.

ChIP sequencing or chromatin immunoprecipitation followed by sequencing (ChIP-seq), is the use of NGS to study protein-DNA interactions in the nucleus. Chromatin immunopercipitation has been used to study such interactions since 1988 (Solomon et al., 1988). It was later used in combination with hybridizaiton microarrays, in a technique called ChIP-chip, and was first used in tandem with NGS in 2007 (Johnson et al., 2007; Barski et al., 2007; Robertson et al., 2007; Mikkelsen et al., 2007). In ChIP, DNA and its bound proteins are cross-linked. Then the DNA-protein complexes are sheared, and protein-specific antibodies are used to select the DNA fragments associated with the protein of interest. The DNA fragments are sequenced and aligned to a reference genome, which enables genome-wide mapping of a protein's activity on the genome at base pair resolution. In spite of the presence of artifacts, ChIP-seq is the gold standard for mapping genome-wide protein-DNA interac-

tions (Wreczycka et al., 2019). The data can be represented as a signal across genomic coordinates, where each base is associated with a *binding strength* real-valued signal (a normalized count of the reads covering that location).

Bisulfite sequencing refers to DNA sequencing preceded by bisulfite treatment of the DNA, a reaction which converts cytosine residues to uracil, unless those cytosines are methylated (Frommer et al., 1992). When the resulting sequence is compared with a reference, an overview of the DNA methylation (methylome), an important epigenetic mark, may be inferred. Bisulfite-treated DNA sequencing (BS-seq) is the gold standard for probing DNA methylation (the methylome) genome-wide (Wreczycka et al., 2017). The methylome is typically characterized by picking CpG's (cytosines followed by guanines) in the genome and measuring how often they are methylated. This is called a beta value. The data can then be represented as an $N_{\text{CpGs}} \times N_{\text{samples}}$ matrix C where c_{ij} is the beta value of CpG i in sample j.

RNA sequencing is used in the study and quantification of all RNA transcripts in a sample. Prior to the development of NGS, hybridization arrays were used to measure gene expression, or the relative abundance of different mRNA transcripts in cells (DeRisi et al., 1996). NGS was first used to quantify gene expression by sequencing reverse-transcribed cDNA in 2008 (Morin et al., 2008). Today RNA sequencing (RNA-seq) has all but replaced microarrays for gene expression quantification, thanks to numerous advantages: it is less susceptible to cross-hybridization mistakes, it has a better dynamic range, offering better detection of highly and lowly expressed genes, and importantly, it does not require the transcript sequence to be known a-priori (Grabherr et al., 2011), and can also be used to identify genetic variants. Using reproducible data analysis pipelines (Wurmus et al., 2018), RNA-seq also provides

superior fidelity than microarrays (Zhao et al., 2014). RNA-seq is performed by sequencing cDNA reverse-transcribed from RNA extracted from a sample. The reads from the sequencing experiment are then aligned to a reference sequence, and each is assigned to the gene (or transcript) it is thought to have originated from. The final product is a read count for each transcript. This is typically characterized as an $N_{\text{transcripts}} \times N_{\text{samples}}$ matrix E, where e_{ij} is some normalized *gene expression* value for transcript i in sample j, which is derived from the read count, typically by normalizing to the library size (number of reads in the sample), the variability of the transcript across samples, and transformed (using the 2-logarithm) so that it follows a normal distribution.

ATAC sequencing (Assay for Transposase Accessible Chromatin) is an assay to determine chromatin accessibility in a sample (Buenrostro et al., 2013). It is performed by first treating DNA with a special transposase which cuts loose accessible chunks of DNA and ligates primers to them. Then, the resulting DNA fragments with primers are amplified and sequenced. When mapped to a reference genome, these show which parts of the genome are accessible, as there will be no reads from compacted chromatin. The accessibility of a genomic segment containing genes is a requirement for polymerase to be able to transcribe any genes from that segment, and so chromatin accessibility is an important epigenetic mark. ATAC sequencing reads which are mapped to the genome may be segmented into "peaks" (segments with many mapped reads), and the data may be represented as an $N_{\text{peaks}} \times N_{\text{samples}}$ matrix T, where t_{ij} is the normalized magnitude of peak i in sample j. These may also be binarized, i.e. $t_{ij} \in \{0, 1\}$.

Other common omics assays which will not be covered in much detail in this disserta-

tion include proteomics and metabolomics, the characterization of the proteins or the profiling of metabolites, respectively. While different techniques exist for both purposes, both proteomics and metabolomics is typically performed using mass spectrometry, thanks to the ability to profile hundreds of thousands of kinds of molecules in parallel, as with high throughput sequencing.

1.3 Single cell RNA sequencing: challenges with single cell transcriptomics

Soon after NGS was applied to bulk transcriptome profiling, it was also used successfully on mRNA extracted from single cells (Tang et al., 2009). Single cell gene expression was done using qPCR (Eberwine et al., 1992) and single-molecule FISH (Tyagi and Kramer, 1996) over a decade earlier. Full-scale transcriptomics using microarrays was first demonstrated in 2003 (Tietjen et al., 2003). However, it was later developments in barcoding and multiplexing (Islam et al., 2011) followed by the use of microfluidics (Klein et al., 2015; Macosko et al., 2015) which really necessitated new computational techniques for single cell RNA sequencing (scRNA-seq). Barcodes, multiplexing, and microfluidics have made it possible to sequence millions of single cells. At the time of writing, the largest study known to me has sequenced the transcriptomes of 1.3 million single cells (10x Genomics, 2017), and the Human Cell Atlas (Regev et al., 2018) has its sights set on *all the cells*. However the small amount of mRNA which can be extracted from a single cell, along with the stochastic nature of transcription, makes scRNA-seq analysis distinct from bulk RNA-seq. After reads from scRNA-seq are mapped to a reference transcriptome, normalization strategies from bulk RNA-seq turned out to be inadequate, as they assume a constant amount

of mRNA per sample, and single cells from different populations can have varying sizes, and hence mRNA content. Normalization, clustering, and differential expression analyses are all affected by the difference in mRNA content between cell populations in ways that bulk RNA-seq experiments are not (reviewed in Bacher and Kendziorski (2016)). In addition, novel analyses have been devised which also take advantage of measurements of other cells when inferring something about a single cell's state. One such ground-breaking technique is *pseudo-temporal ordering* of single cells (Trapnell et al., 2014), which has enabled the study of e.g. cell differentiation trajectories in the transcriptome, while taking fewer temporal time points than would otherwise be neccesary. Studies ranging from cell type identification and discvery, to cell differentiation, cell cycles, and tumor heterogeneity have all benefited greatly from the continued improvements in the number of single cells which may be sequenced and the depth at which they may be sequenced. Whether single cells are manually picked with surgical precision pipetting, or whether sophisticated microfluidics platforms are used to prep hundreds of thousands of single cells, in principle any of the assays described above can be performed on single cells, with the caveat that some of the analytes extracted from single cells (DNA, RNA, proteins) is liable to be lost in the process. Hence, single cell omics stands as an exciting application for all of the above mentioned technologies.

The most common single cell assay at this time is single cell RNA sequencing. A major difficulty with single cell RNA sequencing arrises from the fact that up to 85% of the RNA in a cell may be lost when handling it, due to the physical difficulty of handling such small amounts. In addition, transcription dynamics may result in a cell containing no mRNA transcripts of a gene which is active in that cell at that time in some other sense, e.g. the protein might still be found in the cell*. The resulting data exhibits even stronger bias and high

*In other words, mRNA counts from RNA-seq experiments are only a proxy for gene expression. More

variance than bulk sequencing data, and in addition suffers from zero inflation (drop-outs) which happens when a transcript isn't detected in RNA sequencing although it was present in a cell in some nonzero amount. Many common down-stream analyses for gene expression data are heavily impacted by missing values, and so computational biologists have been imputing missing gene expression data since the days of microarrays (Troyanskaya et al., 2001). Bulk RNA-seq does not suffer from missing values as much as microarrays. However, the drop-out problem in scRNA-seq has lead to a renewed interest in imputation methods specific to scRNA-seq.

Data imputation for single cell RNA sequencing

While estimated transcript abundances in each single cell from a single cell RNA sequencing experiment might only cover about 15% of the total RNA in the cell, microfluidics platforms have made experiments including tens of thousands of cells common. Under the assumption that cells of the same cell type express roughly the same genes with similar relative abundances, it is possible to attempt to *impute*, i.e. fill inn missing values for genes with zero counts, by comparing each single cell to other single cells which are similar to it. Two similar * cells of the same cell type might express a very similar mixture of genes at a given time, but the stochastic transcription, extraction, amplification, and sequencing process might lead to different genes being dropped out from each cell. When the observed part of the two cells' transciptomes are sufficiently similar, we might use them to impute each other's missing values (drop-out genes). Several algorithms have been published that attempt this, with varying levels of sophistication:

on this in chapter 2

*Similar here means the observed part of the transcriptome is similar.

Imputation using mean values The simplest way to impute a missing value is by the mean among non-missing values, i.e. for each observed zero, we may fill in the mean among the nonzero samples. This naïve approach has a major weakness in that it assumes all genes are equally likely to be drop-outs, and thus fills in all zero values. A noteworthy result about the drop-out rate of a gene, i.e. the proportion of cells which have a 0 count of that gene, is that it is exponentially proportional to the mean expression level of that gene (Kharchenko et al., 2014a). Thus, once each gene's mean expression value (across samples) is calculated, each zero count may be filled in to the mean expression *among genes with a similar drop-out rate* to the gene in question. A weakness of this slightly less naïve method is that we expect mean expression of a gene to vary across cell types, and so for experiments profiling more than a single cell type (most single cell RNA sequencing experiments) this is a major weakness. This weakness may be addressed by first clustering cells, and then performing this step within each cluster separately. A representative of this class of imputation methods is CIDR (Lin et al., 2017).

Model based imputation The next step in sophistication is trying to model genes' drop-out likelihood explicitly, as well as genes' expression values as predicted by other genes' expression values. A representative of this class of methods is scImpute (Li and Li, 2017). scImpute first estimates each gene's drop-out likelihood by fitting a bi-modal distribution to the global expression pattern, and assigning for each gene, in each cell, a likelihood that the value is a drop-out as opposed to a true 0 expression. It then uses linear models based on the genes which are predicted to not be drop-outs, to regress the expression values of genes which are determined to be likely drop-outs.

Imputation using local neighborhoods Local neighborhood imputation is based on the

assumption that, in spite of drop-outs, the gene expression profiles of single cells will fall on a smooth manifold where nearby cells are likely to have similar true expression patterns. MAGIC (van Dijk et al., 2017) is such a method, and it works by learning such a manifold using network diffusion on a first-order similarity graph, followed by local averaging i.e. squeezing of the expression profiles to lie exactly on the manifold it predicts. It is the most aggressive of the imputation methods described in this section, changing expression values of all genes in all samples to best explain the observed neighborhoods.

In Chapter 2, I will describe a method I developed for dealing with missing or noisy values in scRNA-seq experiments, and which does not rely on similar cells from the same experiment for the imputation, thus side-stepping the risk of amplifying inherent biases in a single experiment.

1.4 Multi-omics data integration

ome and *omics* refer to comprehensive data types. Multi-omics in turn refers to data spanning different genomics assays. When multiple omics platforms are used on the same biological samples, the resulting multi-modal datasets allow us to probe biological processes from multiple aspects. This is referred to as multi-omics data integration, and is common in the field of cancer research (Hoadley et al., 2014). Multi-omics experiments are also becoming more common in single cells (Dey et al., 2015). In this section, I will discuss some multi-omics analysis strategies, with emphasis on latent variable models for multi-omics integration, focusing, among these, on matrix factorization methods.

The contents of this section are largely reproduced from *Computational Genomics with*

R, Altuna Akalin, Vedran Franke, Jonathan Ronen, Bora Uyar (in press), where I wrote the corresponding chapter. It has been edited for clarity in this dissertation.

Multi-omics data analysis strategies

Like in many other data analytics and machine learning applications, multi-omics data integration algorithms may be split into supervised and unsupervised methods. Unsupervised multi-omics integration methods are methods that look for patterns within and across data types, in a label-agnostic fashion, i.e. without knowledge of the identity or label of the analyzed samples (e.g. cell type, tumor/normal). Supervised integration methods, unlike the unsupervised variety, make use of available labels, such as phenotypes.

Beyond the supervised / unsupervised dichotomy, common strategies for multi-omics data analysis can be further divided into three categories:

Sequential Analysis of multi-omics data includes methods such as CNAmet (Louhimo and Hautaniemi, 2011), where gene expression data (RNA-seq) is used to find up- or down-regulated genes, DNA methylation data (BS-seq) is used to find hyper- and hypo-methylated genes, and copy-number variation data (eg. from DNA-seq) is used to see if genes have gains or losses in DNA copy-number. By analyzing each data type independently, CNAmet computes a score that estimates why genes are up- or down-regulated in cancer; if it is because of aberrant DNA methylation, gain or loss of copy-number, or both. Thus, data from different genomic assays can be used together to study gene expression regulation.

Network-based methods seek to leverage algorithms developed for graph data in order to integrate multi-omics data. They broadly fall into two categories — one where the

network is defined over genes, with edges denoting some kind of interaction or similarity between them, and the other where the network is defined over samples, where edges signify similarity among the samples. In the former (gene-network), the goal is often to identify genes that are implicated in a process, often using the ”guilt through association” method, where a gene network is constructed from multi-omics data, and genes are implicated in some underlying process if they are adjacent to other genes which are related to that process. In the second appraoch (sample network), one builds a graph over samples, where edges signify that samples are close, or similar, in some omics data type. The graph may be trimmed so that only nodes with edges in multiple omics data types retain their edges, and the resulting graph can be used for graph-based analysis such as spectral clustering etc.

Matrix factorization methods have long been a workhorse of unsupervised learning due to their scalability and applicability. The extension from single-data types to multi-omics integration is straightforward. They will be discussed in more detail throughout the rest of this section.

Latent variable models for multi-omics integration

Latent variable models are a dimensionality reduction technique. They make an assumption that the high dimensional data we observe (e.g. counts of tens of thousands of mRNA molecules) arrise from a lower dimension description. The variables in that lower dimensional description are termed *latent variables*, as they are believed to be latent in the data, but not directly observable through experimentation. Therefore, there is a need for methods to infer the latent variables from the data. For instance, the relative abundance of

different mRNA molecules in a cell might be largely determined by the cell type or state. There are other experiments which may be used to discern the cell type (e.g. looking at cells under a microscope), but an RNA-seq experiment does not, directly, reveal whether the analyzed sample was taken from one organ or another. A latent variable model might set the cell type as a latent variable, and the observable abundance of mRNA molecules to be dependent on the value of the latent variable (e.g. if the latent variable is "Regulatory T-cell", we would expect to find high expression of CD4, FOXP3, and CD25).

Matrix factorization for unsupervised multi-omics data integration

Matrix factorization techniques attempt to infer a set of latent variables from the data by finding factors of a data matrix. Principal Component Analysis is a form of matrix factorization which finds factors based on the covariance structure of the data. Generally, matrix factorization methods may be formulated as

$$X = WH,$$

where X is the $N_{\text{features}} \times N_{\text{samples}}$ data matrix, W is an $N_{\text{features}} \times N_{\text{latent factors}}$ feature weight matrix, and H is the $N_{\text{latent factors}} \times N_{\text{samples}}$ latent variable coefficient matrix. This H is the reduced dimension representation. Tying this back to PCA, where $X = U\Sigma V^T$, we may formulate the factorization in the same terms by setting $W = U\Sigma$ and $H = V^T$. If $N_{\text{latent factors}} = rank(X)$, this factorization is lossless, i.e. $X = WH$. However if we choose $N_{\text{latent factors}} < rank(X)$, the factorization is lossy, i.e. $X \approx WH$. In that case, matrix factorization methods normally opt to minimize the error

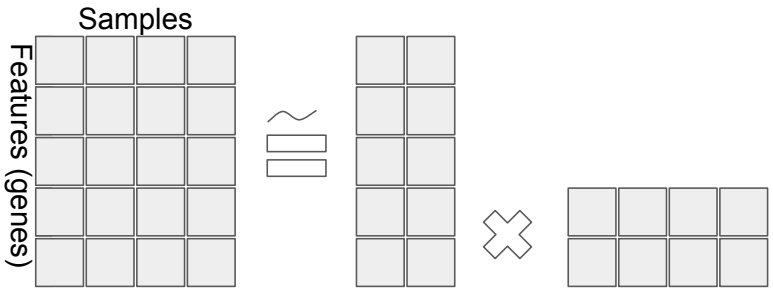


Figure 1.1: General matrix factorization framework. The data matrix on the left hand side is decomposed into factors on the right hand side. The equality may be an approximation as some matrix factorization methods are lossless (exact), while others are an approximation. This figure is reproduced from Akalin et al. (2019).

$$\texttt{min } \|X - WH\|,$$

which may be further subject to some constraints or regularization terms. As we normally seek a latent variable model with a considerably lower dimensionality than X, this is the more common case.

Figure 1.1 illustrates matrix factorization. There, the 5×4 data matrix X is decomposed to a 2-dimensional latent variable model.

Multiple factor analysis (**MFA**) (**Blasius, 2006**) is a natural starting point for a discussion about matrix factorization methods for intergrating multiple data types. It is a straightforward extension of PCA into the domain of multiple data types [*].

Consider Figure 1.2 on the next page, a naïve extension of PCA to a multi-omics con-

[*] When dealing with categorical variables, MFA uses MCA (Multipe Correspondence Analysis). This is less relevant to biological data analysis and will not be discussed here

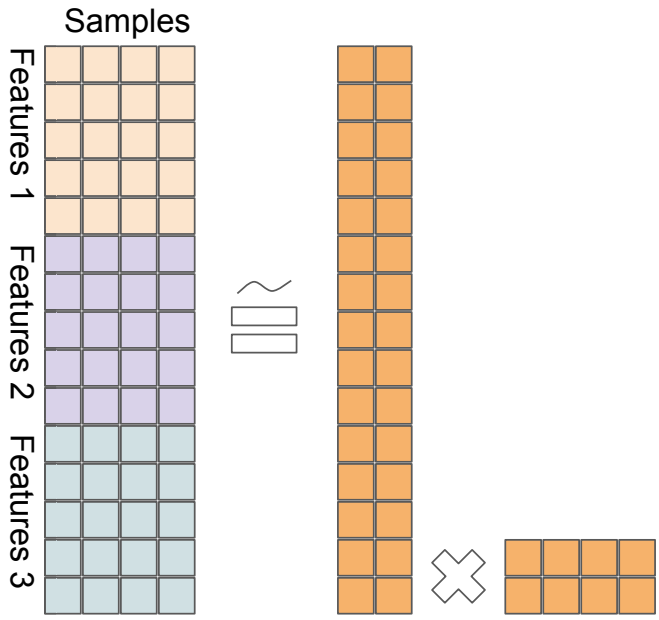


Figure 1.2: A naïve extension of PCA to multi-omics; data matrices from different platforms are stacked, before applying PCA. This figure is reproduced from Akalin et al. (2019).

text. Formally, we have

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix} = WH,$$

a joint decomposition of the different data matrices (X_i) into the factor matrix W and the latent variable matrix H. This way, we can leverage the ability of PCA to find the highest variance decomposition of the data, when the data consists of different omics types. As a reminder, PCA finds the linear combinations of the features which, when the data is projected onto them, preserve the most variance of any K dimensional space. But because measurements from different experiments have different scales, they will also have variance (and co-variance) at different scales.

MFA addresses this issue and achieves balance among the data types by normalizing each of the data types, before stacking them and passing them on to PCA. Formally, MFA is given by

$$\tilde{X} = \begin{bmatrix} X_1/\lambda_1^{(1)} \\ X_2/\lambda_1^{(2)} \\ \vdots \\ X_L/\lambda_1^{(L)} \end{bmatrix} = WH,$$

where X_i are data matrices from different omics platforms, and $\lambda_1^{(i)}$ is the first eigenvalue of the principal component decomposition of X_i, i.e. MFA is equivalent to performing PCA on a concatenated data matrix where each component data matrix is normalized by its largest eigenvalue.

Joint Non-negative Matrix Factorization (Yang and Michailidis, 2015) NMF (Non-negative Matrix Factorization) is an algorithm from 2000 that seeks to find a non-negative additive decomposition for a non-negative data matrix. It takes the familiar form $X \approx WH$, but with the non-negativity constraints $X \geq 0, W \geq 0$, and $H \geq 0$. The non-negative constraints make a lossless decomposition (i.e. $X = WH$) generally impossible. However, many data types are inherently non-negative; for instance, transcript counts from RNA-seq experiments are non-negative, making negative loadings in e.g. PCA undesirable for the purpose of biological interpretation. Hence, NMF attempts to find a solution which minimizes the Frobenius norm of the reconstruction:

$$
\begin{aligned}
min \, \|X - WH\|_F \\
W \geq 0, \\
H \geq 0.
\end{aligned}
$$

This is typically solved for W and H using random initializations followed by iterations of a multiplicative update rule:

$$W_{t+1} = W_t^T \frac{XH_t^T}{XH_tH_t^T} \tag{1.1}$$

$$H_{t+1} = H_t \frac{W_t^T X}{W_t^T W_t X}. \tag{1.2}$$

Since this algorithm is guaranteed only to converge to a local minima, it is typically run several times with random initializations, and the best result is kept.

In the multi-omics context, we will, as in the MFA case, wish to find a decomposition for an integrated data matrix of the form

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix},$$

with X_is denoting data from different geomic assays.

As NMF seeks to minimize the reconstruction error $\|X - WH\|_F$, some care needs

to be taken with regards to data normalization. Different omics platforms may produce data with different scales (i.e. real-valued gene expression quantification, binary mutation data, etc.), and so will have different baseline Frobenius norms. To address this, when doing Joint NMF, we first feature-normalize each data matrix, and then normalize by the Frobenius norm of the data matrix. Formally, we run NMF on

$$\tilde{X} = \begin{bmatrix} X_1^N/\alpha_1 \\ X_2^N/\alpha_2 \\ \vdots \\ X_L^N/\alpha_L \end{bmatrix},$$

where X_i^N is the feature-normalized data matrix $X_i^N = \frac{x^{ij}}{\sum_j x^{ij}}$, and $\alpha_i = \|X_i^N\|_F$.

Another consideration with NMF is the non-negativity constraint. Different omics data types may have negative values, for instance, copy-number variations (CNVs) may be positive, indicating gains, or negative, indicating losses. In order to turn such data into a non-negative form, we will split each feature into two features, one new feature holding all the non-negative values of the original feature, and another feature holding the absolute value of the negative ones. By representing each segment by two rows, one for losses and one for gains, we can represent CNV data as non-negative.

iCluster (Shen et al., 2012) takes a Bayesian approach to the latent variable model. In Bayesian statistics, we infer distributions over model parameters, rather than finding a maximum-likelihood parameter estimates. In iCluster, we model the data as

$$X_{(i)} = W_{(i)}Z + \epsilon_i,$$

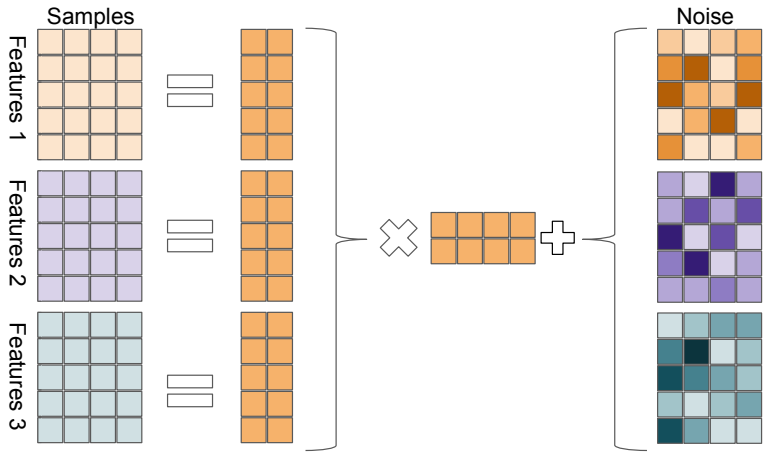


Figure 1.3: Sketch of iCluster model. Each omics datatype is decomposed to a coefficient matrix and a shared latent variable matrix, plus noise. This figure is reproduced from Akalin et al. (2019).

where $X_{(i)}$ is a data matrix from a single genomic assay, $W_{(i)}$ are model parameters, and Z is the latent variable matrix, which is shared between the different omics platforms. ϵ_i is a "noise" random variable, $\epsilon \sim N(0, \Psi)$, where $\Psi = diag(\psi_1, \ldots \psi_M)$ is a diagonal covariance matrix. This is a way of formalizing uncertainty in the model and allow for discrepancies.

With this formulation, the omics measurements X are expected to be the same for samples with the same latent variable representation, up to gaussian noise. Further, we assume a Gaussian prior distribution on the latent variables $Z \sim N(0, I)$, which means we assume $X_{(i)} \sim N\big(0, W_{(i)}W_{(i)}^T + \Psi_{(i)}\big)$. In order to find suitable values for W, Z, and Ψ, we can write down the multivariate normal log-likelihood function and optimize it. For a multivariate normal distribution with mean 0 and covariance Σ, the log-likelihood function is given by

$$\ell = -\frac{1}{2}\Big(\ln(|\Sigma|) + X^T\Sigma^{-1}X + k\ln(2\pi) \Big)$$

(this is simply the log of the Probability Density Function of a multivariate gaussian). For the multi-omics iCluster case, we have $X = \left(X_{(1)}, \ldots, X_{(L)}\right)^T, W = \left(W_{(1)}, \ldots, W_{(L)}\right)^T$, where X is a multivariate normal with o-mean and $\Sigma = WW^T + \Psi$ covariance. Hence, the log-likelihood function for the iCluster modle is given by:

$$\ell_{iC}(W, \Sigma) = -\frac{1}{2}\left(\sum_{i=1}^{L} \ln(|\Sigma|) + X^T\Sigma^{-1}X + p_i \ln(2\pi)\right)$$

where p_i is the number of features in omics data type i. Because this model has more parameters than we typically have samples, we need to push the model to use fewer parameters than it has at its disposal, by using regularization. iCluster uses Lasso regularization, which is a direct penalty on the absolute value of the parameters. I.e., instead of optimizing $\ell_{iC}(W, \Sigma)$, we will optimize the regularized log-likelihood:

$$\ell = \ell_{iC}(W, \Sigma) - \lambda\|W\|_1.$$

The parameter λ acts as a dial to weigh the tradeoff between better model fit (higher log-likelihood) and a sparser model, with more w_{ij}s set to 0, which gives models which may generalize better and are more interpretable.

In order to solve this optimization problem, iCluster employs the Expectation Maximization (EM) algorithm (Dempster et al., 1977), the full details of which are beyond the scope of this text. I will introduce a short sketch instead. The intuition behind the EM algorithm is a more general case of the k-means clustering algorithm.

EM algorithm sketch

- Initialize W and Ψ

- Until convergence of W, Ψ
  - E-step: calculate the expected value of Z given the current estimates of W and Ψ and the data X
  - M-step: calculate maximum likelihood estimates for the parameters W and Ψ based on the current estimate of Z and the data X.

iCluster+ (Mo and Shen, 2013) is an extension of the iCluster framework, which allows for omics types to arrise from other distributions than a gaussian. While normal distributions are a good assumption for log-transformed, centered gene expression data, it is a poor model for binary mutations data, or for copy number variation data, which can typically take the values $(-2, 1, 0, 1, 2)$ for heterozygous / monozygous deletions or amplifications. iCluster+ allows the different Xs to have different distributions:

- For binary mutations, X is drawn from a multivariate binomial,
- for normal, continuous data, X is drawn from a multivariate gaussian,
- for copy number variations, X is drawn from a multinomial,
- for count data, X is drawn from a poisson.

In that way, iCluster+ allows us to explicitly model our assumptions about the distributions of our different omics data types, and leverage the strengths of Bayesian inferrence.

Both iCluster and iCluster+ make use of sophisticated Bayesian inference algorithms (EM for iCluster, Metropolis-Hastings MCMC for iCluster+), which means they do not scale up trivially. Therefore, it is recommended to filter down the features to a

manageable size before inputing data to the algorithm. The exact size of "manageable" data depends on your hardware, but a rule of thumb is that dimensions in the thousands are ok, but in the tens of thousands might be too slow.

Multi-Omics Factor Analysis (MOFA) (Argelaguet et al., 2018) is a latent factor multi-omics integration method which solves the same problem definition as iCluster and iCluster+ (Figure 1.3 on page 25). However, while iCluster solves the poblem using the EM algorithm, and iCluster+ using MCMC, MOFA uses variational mean-field approximation. A benchmark comparing MOFA to iCluster+, as well as to *maui*, a method I developed, is presented in Chapter 3.

Down-stream analysis for latent variable methods

The latent variable methods described above can only ever make up part of an interesting analysis of multi-omics data. However, a latent factor representation of multi-modal data facilitates down-stream analysis which would be difficult to do without first reducing the dimensionality of the data. This section describes common analyses which are assisted by latent factor methods.

Clustering using latent factors

A common analysis in biological investigations is clustering. This is often interesting in cancer studies as one hopes to find groups of tumors (clusters) which behave similarly, i.e. belong to the same risk group and/or respond to the same drugs. In single cell studies, clustering is often used to define cell types. Additionally, clustering is often used in studies which aim to find differences between e.g. gene expression of some treatment sample and

a control, where clustering may be seen as a confirmatory step that shows treatment and control groups are indeed discriminable in experimental data. PCA is a common step in clustering analyses, and so it is easy to see how the latent variable models discussed above may all be a useful pre-processing step before clustering.

One-hot clustering A specific clustering method which is used in tandem with NMF is to assume each sample is driven by one component, i.e. that the number of clusters K is the same as the number of latent variables in the model and that each sample may be associated to one of those components. We assign each sample a cluster label based on the latent variable which affects it the most.

The one-hot clustering method does not lend itself very well to the other methods discussed above, i.e. iCluster and MFA. The latent variables produced by those other methods may be negative, and further, in the case of iCluster, are going to assume a multivariate gaussian shape. As such, it is not trivial to pick one ”dominant factor” for them. For NMF variants however, this is a very common way to assign clusters.

K-means clustering is a special case of the EM algorithm, and indeed iCluster was originally motivated as an extension of K-means from binary cluster assignments to real-valued latent variables. The iCluster algorithm, as it is so named, calls for application of K-means clustering on its latent variables, after the inference step. K-means lends itself well to clustering on latent factors which are normally distributed, as the main assumption of K-means clustering is that each sample will be closer to the mean of its cluster than to that of other clusters.

Biological interpretation of latent factors

Latent factor representations are useful for down-stream analysis such as clustering, but biologists are rightfully averse of so-called *black box* models, where the relationship between what goes in (omics) and what comes out (latent factors) is opaque. Hence, it is important that we are able to *interpret* the meaning of latent factors in the context of what is known about the biology of the features (e.g. genes) they summarize.

Inspection of feature weights in loading vectors The most straightforward way to go about interpreting the latent factors in a biological context, is to look at the coefficients which are associated with them. The latent variable models introduced above all take the linear form $X \approx WH$, where W is a factor matrix, with coefficients tying each latent variable with each of the features in the L original multi-omics data matrices. By inspecting these coefficients, we can get a sense of which multi-omics features are co-regulated.

Disentangled representations A desirable property of latent factor representations, one which simplifies their interpretation and down-stream analysis, is for each input feature to be prediminantly associated with a single latent factor. This property is termed *disentangledness*, i.e., it leads to *disentangled* latent variable representations, as changing one input feature only affects a single latent variable.

Enrichment analysis The recent decades of genomics have uncovered much about the ways in which genes cooperate to perform biological functions in concert. This work has resulted in rich annotations of genes, groups of genes, and the different functions they carry out. Examples of such annotations include the Gene Ontology Consortium's *GO terms* (Ashburner et al., 2000; Consortium, 2017), the *Reactome pathways*

database (Fabregat et al., 2018), and the *Kyoto Encyclopaedia of Genes and Genomes* (Kanehisa et al., 2017). These resources, as well as others, publish lists of so-called *gene sets*, or *pathways*, which are sets of genes which are known to operate together in some biological function, e.g. protein synthesis, DNA mismatch repair, cellular adhesion, and many, *many* other functions.

These gene sets are used to annotate genes of interest in particular studies. For instance, by performing a differential expression analysis we migh uncover a set of genes which are upregulated under some condition. By comparing these genes with the published gene sets, we may find that similar genes are implicated in other biological processes. This is termed gene set enrichment analysis, and is commonly used by researchers to further their understanding of the systems they study.

In the context of making sense of latent factors, the question we will be asking is whether the genes which drive the value of a latent factor (the genes with the highest factor coefficients) also belong to any interesting annotated gene sets, and whether the overlap is greater than we would expect by chance. If there are N genes in total, K of which belong to a gene set, the probability that k out of the n genes associated with a latent factor are also associated with a gene set is given by the hypergeometric distribution: $P(k) = \frac{\binom{K}{k}-\binom{N-K}{n-k}}{\binom{N}{n}}$.

The *hypergeometric test* uses the hypergeometric distribution to assess the statistical significance of the presence of genes belonging to a gene set in the latent factor. The null hypothesis is that there is no relationship between genes in a gene set, and genes in a latent factor. When testing for over-representation of gene set genes in a latent factor, the P value from the hypergeometric test is the probability of getting k or more genes from a gene set in a latent factor: $p = \sum_{i=k}^{K} P(k = i)$.

The hypergeometric enrichment test is also referred to as *Fisher's one-sided exact test*. This way, we can determine if the genes associated with a factor significantly overlap (beyond chance) the genes involved in a biological process. Because we will typically be testing many gene sets, we will also need to apply multiple testing correction, such as Benjamini-Hochberg correction.

The remainder of this thesis

In the chapters that follow, I will describe in depth two novel methods I have developed as part of my doctoral studies. The first method, called *netSmooth* (Ronen and Akalin, 2018a), uses priors from earlier published data in order to temper noisy measurements in scRNA-seq. In chapter 2, I will expand on ideas introduced on page 14, and introduce the *netSmooth* method. I will show results comparing *netSmooth* to other state-of-the-art pre-processing methods for scRNA-seq on different scRNA-seq datasets representing different biological systems.

In chapter 3, I will describe a deep learning based method I developed called *maui* (Ronen et al., 2018), which learns latent factor representations of multi-omics data using a stacked variational autoencoder. While it is applicable to any multi-omics dataset, be it from single cells or bulk, I will demonstrate its utility in sub-typing colorectal cancers using bulk multi-omics data from primary tumor biopsies. In the same chapter, I will also show that *netSmooth* can be repurposed for pre-processing of data other than scRNA-seq, and that using *netSmooth* together with *maui* enables us to further improve the clinical relevance of cancer sub-types.

2

Imputing scRNA-seq with priors from other experiments

Single cell RNA sequencing (scRNA-seq) is a powerful assay which allows researchers to probe genome-wide expression patterns in millions of single cells. However, with measurements from each single cell covering as little as 15% of its transcriptome, the utility of such studies depends on computational data imputation methods. Imputation methods

typically involve guessing an observed 0's true value by looking at other cells which are similar (see section 1.3.1 on page 14). While this approach has demonstrated utility in many cases, it is not without issues. One issue is that single cell RNA sequencing experiments also suffer from technical biases, e.g. different capture rates for different genes. In the presence of such inherent biases in the data from a single experiment, imputing using other datapoints from the same experiment can amplify these technical biases, overshadowing the true biological variability that experimenters are interested in. In this chapter, I describe a different approach to data imputation - one which uses data from *other* experiments as a template.

The contents of this chapter are adapted from Ronen and Akalin (2018a). Most sections are reproduced verbatim, but have been rearranged and edited to fit this format.

2.1 Introduction

With unprecedented throughput and resolution, scRNA-seq has enabled many studies which were previously impractical, such as characterization of cell type heterogeneity, differentiation, and developmental trajectories (Wagner et al., 2016). However, the adaptation of RNA sequencing techniques from bulk samples to single cells did not progress without challenges. Typically, only a fraction of a cell's transcriptome is captured by the experiment, leading to so called *dropout* events where a transcript is not observed in some cell in spite of it being expressed there. RNA-seq experiments produce read counts quantifying the abundance of different transcripts, or genes*, in a cell. When an expressed gene is not captured in an RNA-seq experiment, the resulting count for that gene will be zero; this is what is most commonly referred to as a dropout event. When a fraction of

*I will use transcripts and genes interchangeably throughout this section.

a gene's mRNA is captured by an RNA-seq experiment, producing a read count which is lower than the actual expression level (relative to other genes), it is also referred to as a dropout. The dropout rate is related to the population level expression of a gene, leading to many false zero counts for lowly expressed genes, and artificially low counts for highly expressed ones (Kharchenko et al., 2014b). The dropout rate is also related to the biology of the cell type, as some cell types transcribe fewer genes than do others, which may appear intistinguishable from dropout events (Kharchenko et al., 2014b). To what extent dropouts are a technical artifact, and how much they are caused by "bursty" transcription, remains an open question; but for a range of reasons, only some of which are understood, when summed over many samples, transcript counts from single cells resemble those of bulk experiments (Wu et al., 2014), while across individual cells there is significant variation. This makes analysis more difficult than in bulk RNA sequencing experiments.

Computational methods designed to deal with these issues treat dropout events as missing data points, whose values are to be imputed based on non-missing data points (observed measurements). The proportion of cells with zero counts per gene, a proxy for its technical dropout rate, is a function of the population-wise mean expression of that gene (Pierson and Yau, 2015; Kharchenko et al., 2014b). This observation has led researchers to treat zero counts as dropout candidates to be imputed.

CIDR (Lin et al., 2017) attempts to impute missing values based on the predicted mean expression of a gene, given its empirical dropout rate (zero count). scImpute (Li and Li, 2017) estimates dropout likelihoods per gene and per sample, and assigns each gene in each sample a status as a dropout candidate. scImpute may consider genes to be likely dropouts even with nonzero expression, and zero count genes might not be considered likely dropouts, based on their population-wide expression distributions. scImpute then uses a regularized linear model to predict the expression of dropout genes based on the expression of likely

non-dropouts in all other cells. MAGIC (van Dijk et al., 2017) performs local averaging after building a topological graph of the data, updating the expression value of all genes in all cells to their local neighborhood average.

The methods mentioned above use information present in the data in order to impute the missing information within the same data. As such, they amplify whatever biases are present in a dataset; similar cells pre-imputation will become more similar after imputation, as expression profiles of non-dropout genes will drive similarities in imputed dropped-out genes. Further, all methods except MAGIC only impute unobserved expression events (zeros or near zeros), while in actuality, the dropout phenomenon affects the whole transcriptome. Hence, imputation methods for scRNA-seq should also adjust non-zero expression measurements in order to recover the true signal.

In the following sections, I will present a method I developed, called *netSmooth*, that uses prior knowledge to temper noisy experimental data. RNA sequencing experiments produce transcript count data as a proxy for gene activity, which is not known a-priori, especially for experiments profiling unknown cell types. However, decades of molecular biology research have revealed much about the principles of gene expression co-regulation. For instance, genes coding for proteins that interact with one another are likely to be co-expressed in cells (Bhardwaj and Lu, 2005; Fraser et al., 2004), and as such, protein-protein interaction (PPI) databases (Szklarczyk et al., 2017; Lee et al., 2011a) describe genes' propensity for co-expression. I developed a method which uses diffusion on a PPI graph to *network-smooth* gene expression values. Each node in the graph (a gene) has an associated gene expression value, and the diffusion represents a weighted averaging of gene expression values among adjacent nodes in the graph, within each cell. This is done iteratively until convergence, strengthening co-expression patterns which are expected to be present. Effectively, this adjusts the observed gene expression values to more closely conform to the prior ex-

pectation of co-expression patterns encoded by the PPI network. As I will show in the following sections, incorporation of prior data from countless experiments in the pre-processing of scRNA-seq experiments improves resistance to noise and dropouts. Similar network based approaches have been used to extract meaningful information from sparse mutational profiles (Hofree et al., 2013; Vandin et al., 2011), and indirectly on gene expression data by diffusing test statistics on the network to discover regulated gene candidates (Dørum et al., 2011). *netSmooth* uses diffusion of gene expression values directly on the PPI network for data denoising and imputation. In the absence of ground-truth, the parameter of this method, the diffusion rate, may be optimized using data driven metrics. I applied *netSmooth* to a variety of single cell experiments and compared its performance to other selected imputation methods, namely scImpute and MAGIC. These methods represent divergent ways of imputing the scRNA-seq data.

While I mention CIDR in this review, I do not include comparisons to CIDR in the main text, alongside MAGIC and scImpute. This is because CIDR uses its own clustering procedure as a part of the imputation workflow. scImpute and MAGIC are agnostic about post-imputation analysis, and therefore more readily lend themselves to comparison with *netSmooth* using a unified analysis framework (see section 2.2). For completeness, the benchmark results of CIDR are included in the supplement (Figure A.11 on page 122, Figure A.16 on page 124).

I also made *netSmooth* available as an R package, so that other researchers may use it on their own data. At the time of writing, it has been downloaded more than 1,500 times[*]. It is available on GitHub: `https://github.com/BIMSBbioinfo/netSmooth` and Bioconductor: `https://bioconductor.org/packages/release/bioc/html/netSmooth.html`.

[*] `http://bioconductor.org/packages/stats/bioc/netSmooth/`

2.2 Methods and Data

In this section, I will describe in detail the *netSmooth* method, as well as down-stream analysis procedures and benchmarks which are implemented in the *netSmooth* R package and will be used later in the results section, including data-driven metrics for optimizing *netSmooth*'s parameter. Then I will describe the construction of the gene-gene network which ships with the *netSmooth* R package, as well as other gene-gene networks constructed for this analysis. Finally, I will describe the data sets that are analyzed in the results section, and the parameter tuning of the comparison methods.

Overview of the method

The intuition behind the *netSmooth* algorithm is that gene networks encoding co-expression patterns can be used to smooth scRNA-seq data, pushing its coexpression patterns in a biologically meaningful direction. Here, I demonstrate this using PPI networks, which are predictive of coexpression (Fraser et al., 2004). I produced a PPI graph of high-confidence interactions based on the PPI database STRINGdb (Szklarczyk et al., 2017), which I will use in the next sections to demonstrate the utility of this method. In a later section, I will also demonstrate the robustness of the method to other sources of gene networks, and its sensitivity to the fidelity of the gene-gene interactions represented in the network.

netSmooth takes two inputs: (1) a gene expression matrix, N genes by M cells, and (2) a graph where genes are nodes, and edges indicate genes which are expected to be co-expressed. The edges may be weighed, indicating the strength or direction of a relationship; an edge weight of 2 indicates stronger expected co-expression (e.g. correlation) than an edge weight

of 1, and an edge weight of -1 indicates negative expected co-expression (e.g. anticorrelation), such as one gene being a repressor for another. The expression profile of each cell is then projected onto the graph, and a diffusion process is used to smooth the expression values, within each sample, of adjacent genes in the graph (Figure 2.1 on the following page). In this way, post-smoothing values of genes represent an estimate of activity levels based on reads aligned to that gene, as well as those aligned to its neighbors in the graph. Thus, a gene with a low read count (possible technical dropout), whose neighbors in the graph are highly expressed, will get a higher value post smoothing. The rate at which expression values of genes diffuse to their neighbors is degree-normalized, so that genes with many edges will affect their neighbors less than genes with more specific interactions. The diffusion is done using a *random walks with restarts* (RWR) process (Vandin et al., 2011), where a conceptual random walker starts in some node in the graph, and at each iteration moves to a neighboring node with a probability determined by the edge weight between the nodes, or, with some probability (the restart rate), restarts the walk from the original node. The *network-smoothed* value is the stationary distribution of this markov process. The RWR process has one free parameter, the restart rate. A low value for the restart rate allows diffusion to reach further in the graph; a high restart rate will lead to more local diffusions.

Network diffusion through random walks

The *netSmooth* algorithm takes a graph $G = \{V, E\}$ where $V = \{gene_i\}$ is the set of genes, and $E = \{(i \rightarrow j)\}$ is the set of edges between genes. The edge weights are degree-normalized, so that each gene's outgoing edges' weights sum to 1. We then define a process of random walk with restarts as in (Vandin et al., 2011), on the PPI graph, where a conceptual random walker starts on a node in the graph (a gene/protein) and at each step

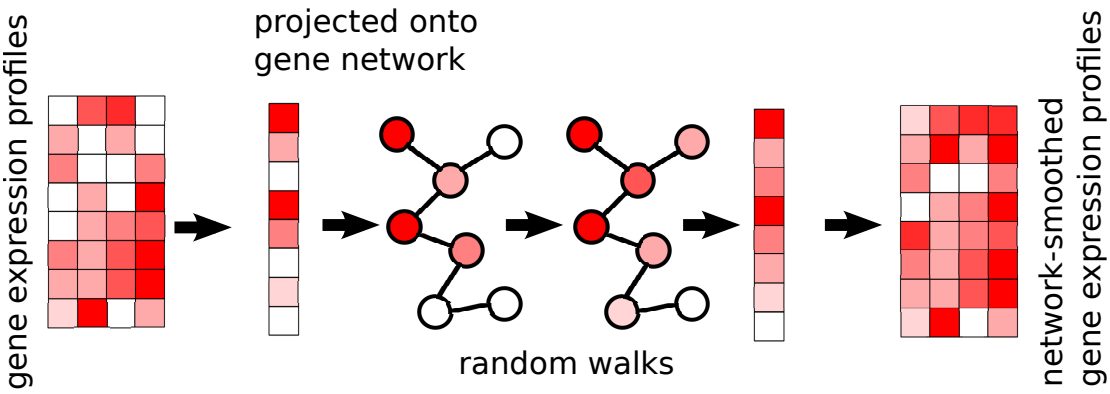


Figure 2.1: The netSmooth algorithm takes a gene expression profile, and a gene network. The expression profile of each sample is projected onto the network, where a diffusion process allows genes' expression values to be smoothed by their neighbors'. This is done for each cell independently of others. The end result is a network smoothed gene expression matrix. This figure is reproduced from Ronen and Akalin (2018a)

walks to an adjacent node with the probability determined by the α times the edge weight. Further, at each step, there is a probability of $(1-\alpha)$ that the walker restarts to its original node. $(1-\alpha)$ is called the *restart rate*. This process is done starting at each node in the graph.

Mathematically, given a graph defined by an adjacency matrix $A_{[MxM]}$, where A_{ij} is the edge weight between gene i and gene j (and o for unconnected genes), and a vector $f_{[Mx1]}$, where f_i^t is the probability that the walker is at node i at step t, the process is defined by

$$f^{t+1} = \alpha \mathbf{A} f^t + (1-\alpha) f^0.$$

This process is convergent, and the stationary distribution is easy to solve for, by setting $f^t = f^{t+1}$:

$$f^{\infty} = (1-\alpha)(I - \alpha \mathbf{A})^{-1} f^0.$$

Hence, the random walk with restarts process is a diffusion process defined on the PPI

graph, or through the diffusion kernel (smoothing kernel)

$$K_A^\alpha = (1-\alpha)(I - \alpha \mathbf{A})^{-1}$$

where $(1-\alpha)$ is the restart probability, and A is the (column normalized) adjacency matrix of the PPI graph. Consequently, we define the *network-smoothed* expression profile

$$E_{sm} = K_A^\alpha E,$$

where $E_{[MxN]}$ is the normalized count values of the M genes in the N cells.

A robust clustering procedure

Clustering analysis features prominently in scRNA-seq analyses; whether recapitulating known results or discovering new cell types, clustering cells by their gene expression profiles is commonly used to identify distinct populations. While some approaches directly take into account the zero-inflation of scRNA-seq data (Lin et al., 2017), other studies use traditional methods (Deng et al., 2014). There is no standard method for clustering single cell RNA-seq data, as different studies produce data with different topologies, which respond differently to the various clustering algorithms.

In order to avoid optimizing different clustering routines for the different datasets I benchmark on, and also in order to avoid optimizing clustering routines to the imputation procedure, I have implemented a robust clustering routine based on *clusterExperiment*[*] (Purdom and Risso, 2017), a framework for robust clustering based on a consensus of dif-

[*]Version 1.4.0, available from Bioconductor https://bioconductor.org/packages/release/bioc/html/clusterExperiment.html

ferent clustering algorithms, different parameters for these algorithms, and different views of the data. The different views are different reduced dimensionality projections of the data based on different techniques. No single clustering result will dominate the data, and only cluster structures which are robust to different analyses will prevail. This procedure will very likely be sub-optimal for each of the tasks we use it on, but this is by design; by averaging over many clustering results, it represents an unbiased, reproducible cluster assignment. The procedure I implemented using the framework is as follows:

1. Perform different dimensionality reduction techniques on the data
    - PCA on the 500 most variable genes
        - with 5 components
        - with 15 components
        - with 50 components
    - Alternatively to PCA, t-SNE on the 500 most variable genes
        - with 2 dimensions
        - with 3 dimensions
    - Select the most variable genes
        - 100 most variable genes
        - 500 most variable genes
        - 1000 most variable genes
2. The PCA, t-SNE, and variable gene subsets make up different views of the data. On each view of the data, perform PAM clustering* with K ranging from 5 to 10.

*Partitioning Around Medioids, or k-medioids clustering, is a relative of k-means clustering, where each cluster is represented by a medioid (an actual example) rather than the mean.

3. Calculate the co-clustering index for each pair of samples. This is the proportion of times the samples are clustered together, in the different clustering results based on the different views and clustering parameters above.

4. Find a consensus clustering from the co-clustering matrix. This is done by constructing a dendrogram using average linkage, and traversing down the tree until a block with a self-similarity of at least 0.6, and a minimum size of 20 samples emerges. (instead of using `cutree`).

5. Perform hierarchical clustering of the cluster medioids, with similarities based on expression of the 500 most variable genes.

6. Perform a DE analysis between clusters that are adjacent in the hierarchy from (5), and merge them if the proportion of genes that are found to be significantly differentially expressed between them (adjP $< .05$) is less than than 0.1.

Using only the 500 most variable genes ensures the biological variation will dominate the technical variation, and enhances the reproducibility of t-SNE (McCarthy et al., 2017).

Importantly, samples that at step (4) don't have a high enough affinity to any emerging cluster, will not be assigned to any cluster. The clustering is performed using the `clusterExperiment::clusterSingle` and `clusterExperiment::clusterMany` functions, the consensus clustering is obtained using the `clusterExperiment::combineMany` function, and the cluster merging (steps 5 and 6) using the `clusterExperiment::makeDendrogram` and `clusterExperiment::mergeClusters` functions. For more details, see Purdom and Risso (2017).

Dimensionality reduction in the clustering procedure

In step (1) above, we cluster cells in a lower dimension embedding using either PCA (Hastie et al., 2001) or t-SNE (van der Maaten and Hinton, 2008), in a dataset-dependent manner. Different scRNA-seq datasets respond better to different dimensionality reduction techniques which are better able to tease out the biological cluster structure of the data. In order to pick the right technique algorithmically, we compute the entropy in a 2D embedding. We obtain 2D embeddings from the 500 most variable genes using either PCA or t-SNE, bin them in a 20x20 grid, and compute the entropy using the `discretize` and `entropy` functions in the *entropy* R package* (Hausser and Strimmer, 2014). The entropy in the 2D embedding is a measure for the information captured by it. For the clustering procedure, we pick the embedding, either PCA or t-SNE, with the highest information content.

Optimizing the smoothing parameters by cluster robustness

The *netSmooth* algorithm, given a gene network, has one free parameter - the restart rate of the random walker, $(1 - \alpha)$. Alternatively, α is the complement of the restart rate. An $\alpha = 0$ indicates a unit restart rate and consequently no smoothing; an $\alpha = 1$ corresponds to a random walk without restarts. Intermediate values for α result in increasing levels of smoothing; the value of α determines how far random walks can go on the graph before restarting, or how far along the network a gene's influence is likely to reach. It is tempting to optimize α with respect to the variable the experiment sets out to measure,

*Version 1.2.1, available from CRAN: https://cran.r-project.org/web/packages/entropy/index.html

e.g. if we set to identify clusters of different cell types, we might be tempted to pick the α which results in the highest cluster purity. However, in many experiments the identity of the samples is not known a-priori. Therefore, I propose a data driven workflow to pick a sensible value for α.

One such data-driven statistic is the proportion of samples assigned to robust clusters; the robust clustering procedure (see page 41) only assigns samples to clusters if they have a strong enough affinity to it. It is able to leave samples without a cluster assignment. I proposed to use the proportion of samples which can be clustered as a metric to optimize using α. For two of the three datasets I will demonstrate below, picking the α that gives the highest proportion of cells in robust clusters, also gives the clusters with the highest purity index (Figure A.2 on page 114). Importantly, this metric is entirely data-driven and does not require external labels, making it feasible for any scRNA-seq study. The results in the next section all use the value of α picked to optimize proportion in robust clusters.

Benchmarks: cluster purity and adjusted mutual information

In order to benchmark the usefulness of clustering results where ground-truth labels are known, I used a cluster purity index, and an adjusted mutual information score. The cluster purity metric refers to the proportion of the samples in a cluster which are of the dominant cell type in that cluster. The purity for cluster i is given by

$$Purity_i = \frac{\sum_{j \in C_i} \begin{cases} 1, & \text{if } label_j = \text{dom}_i \\ 0, & \text{otherwise} \end{cases}}{n_i}$$

where $C_i = \{j | \text{cell}_j \in \text{cluster}_i\}$, $label_j$ is the cell type of $cell_j$, $n_i = |C_i|$ is the number of cells in cluster i, and

$$\text{dom}_i = \arg\max_l \sum_{j \in C_i} \begin{cases} 1, & \text{if } label_j = l \\ 0, & \text{otherwise} \end{cases}$$

is the dominant cell type in cluster C_i.

In addition to the cluster purity metric, I computed the Adjusted Mutual Information (AMI, Vinh et al. (2010)), an information theoretic measure of clustering accuracy which accounts for true positives (two cells of the same type in the same cluster) being caused by chance. The AMI between a clustering C and the true labels L is given by

$$AMI(L, C) = \frac{MI(L, C) - E[MI(L, C)]}{max(H(L), H(C)) - E[MI(L, C)]},$$

where $MI(a, b)$ is the mutual information between labellings a and b, $H(a)$ is entropy of clustering a, and $E[\cdot]$ denotes the expectation.

Construction of the gene-gene network

The PPI graph from which the diffusion kernel was derived was constructed using data from STRINGdb (Szklarczyk et al., 2017). For each pair of proteins, STRINGdb provides a *combined interaction score*, which is a score indicating how confident we can be in the interaction between the proteins, given the different kinds of evidence STRINGdb collates*. We subset the links to only those above the 90th percentile of combined interaction scores,

*STRINGdb collects diverse evidence such as co-expression, biochemical interactions, genetic neighborhoods, and even text mining, where proteins are assigned a likelihood to interact if they appear together frequently in abstracts on PubMed. See https://string-db.org

only keeping the 10% most confident interactions. For mouse that is 1,020,816 interactions among 17013 genes. For human, 852,722 interactions among 17467 genes.

Other gene networks were constructed using HumanNet (Lee et al., 2011a). As the name suggests, this is a functional gene network of Human genes, and so I only used it for the human data (see below). The HumanNet graph isn't as dense as the STRINGdb graph, and so requried no filtering.

The data sets

The results in the next section use data from the following three publically available scRNA-seq datasets. The hematopoiesis dataset (Nestorowa et al., 2016) was obtained from the Gene Expression Omnibus (Edgar et al., 2002). The embryonic (Deng et al., 2014) and glioblastoma (Patel et al., 2014) datasets were obtained from *conquer* (Soneson and Robinson, 2017), a repository of uniformly processed scRNA-seq datasets. The datasets are available, see Table 2.1.

Dataset	URL
Hematopoiesis	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE81682
Embryonic cells	http://imlspenticton.uzh.ch/robinson_lab/conquer/data-mae/GSE45719.rds
Glioblastoma	http://imlspenticton.uzh.ch/robinson_lab/conquer/data-mae/GSE57872.rds

Table 2.1: Datasets and availability. This table is reproduced from Ronen and Akalin (2018a)

MAGIC and scImpute parameters

For all the results presented in section 2.3 on the next page, scImpute was run using the default parameters (`drop_thre = 0.5`). For MAGIC, we used values for the diffusion

time parameter ($T = \{1, 2, 4, 8, 16\}$). Unlike *netSmooth*, for MAGIC the proportion of samples in robust clusters and the cluster purities were anti-correlated; thus we picked the one that gave the best cluster purities as the best MAGIC parameter. The chosen T values are given in Table 2.2. We used MAGIC version 0.1* and scImpute version 0.0.2†.

Dataset	Optimal T
Hematopoiesis	1
Embryonic cells	4
Glioblastoma	2

Table 2.2: Optimal diffusion time values for MAGIC. This table is reproduced from Ronen and Akalin (2018a)

The netSmooth R package

The analysis was done using the *netSmooth* R-package (Ronen and Akalin, 2018b), which is available online: `https://github.com/BIMSBbioinfo/netSmooth`. The *netSmooth* R package was included in the 3.7 release of Bioconductor: `https://bioconductor.org/packages/release/bioc/html/netSmooth.html` and has since been downloaded over 1,500 times. It is available under a GPL version 3 (or later) license.

2.3 Results

In the following pages, I will show the utility of *netSmooth* on three analysis tasks, using three different scRNA-seq datasets studying three different cellular systems: Hematopoiesis, Embryonic development, and cancer. These results were published in Ronen and Akalin (2018a), and are reproduced here with minor edits for clarity.

*Available from GitHub: `https://github.com/pkathail/magic`.
†Available from GitHub: `https://github.com/Vivianstats/scImpute`.

netSmooth improves cell type identification from scRNA-seq

Nestorowa et al. (2016) sequenced the transcriptomes of 1645 mouse hematopoietic stem/progenitor cells (HSPCs), and also assayed them using flow cytometry, FACS-sorting them into 12 common HSPC phenotypes. This presents an atlas of the hematopoiesis process at a single cell resolution, showing the differentiation paths taken by E-SLAM HSCs as they differentiate to E, GM, and L progenitors. The authors of this study demonstrate that upon clustering the data, some clusters corresponds to cell types. However, the clusters are not noise free and do not fully recapitulate cell type identity. We obtained clusterings of the cells from the normalized counts, as well as after application of *netSmooth*, MAGIC (van Dijk et al., 2017), and scImpute (Li and Li, 2017), using a robust clustering procedure based on the *clusterExperiment* R package (Purdom and Risso, 2017) (See Section 2.2 on page 38). After clustering, we used the edgeR-QLF test (Robinson et al., 2010) to identify genes that are differentially expressed in any of the discovered clusters. Figure 2.2a,b on page 50 shows the log-transformed expression values of the 500 most differentially expressed genes, before and after application of *netSmooth*. The column annotations indicate the FACS-sorted cell type of each cell, as well as the cluster assignment obtained from the robust clustering procedure, using the *netSmooth* R package. The figure suggests that the network-smoothing effect is subtle on the individual genes, as difference between the heatmaps is negligible visually. Figure 2.2c,d shows the same for the MAGIC and scImpute-preprocessed data, respectively. MAGIC seems to do the strongest transformation to the data, as is also seen in lower dimension embeddings (Figure A.3 on page 115, Figure A.4 on page 116).

As this dataset has cells with labels independent of the RNA-seq (FACS-sorted phe-

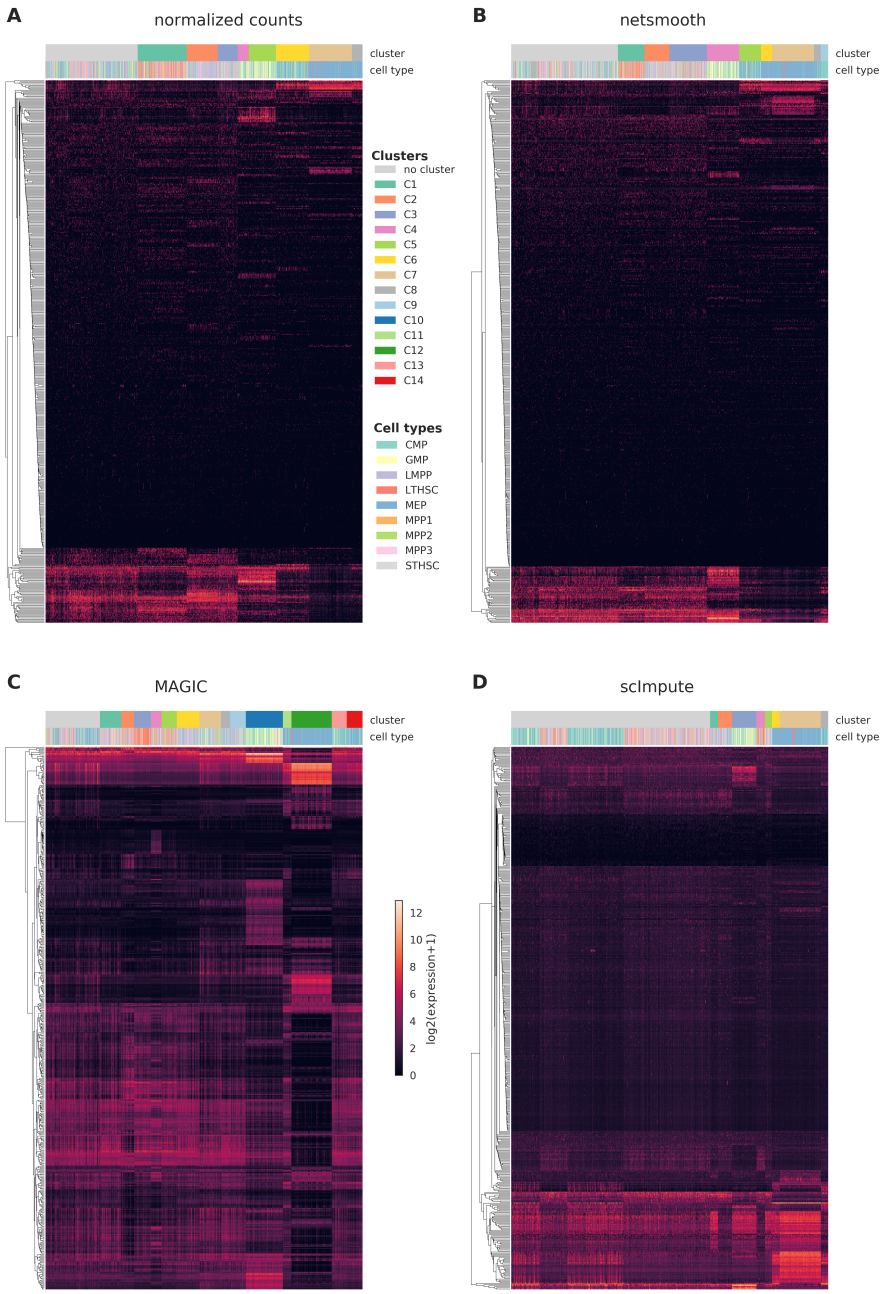


Figure 2.2: Cells were clustered using the robust clustering procedure, and the log-transformed expression values of the 500 most differentially expressed genes (by edgeR-QLF test adjusted P value) in any of the discovered clusters are shown in a heatmap, as well as cluster assignments and FACS-sorted cell types. **A)** raw (no imputation), **B)** after application of netSmooth, **C)** missing values imputed using MAGIC **D)** missing values imputed using scImpute. This figure is reproduced from Ronen and Akalin (2018a).

notypes), it presents us with an opportunity to compare the gene expression levels (as measured by RNA-seq), to a meaningful phenotypic variable, i.e. the cell type. The cell type discrimination of a clustering result is compared using a cluster purity metric and and the adjusted mutual information (AMI). The cluster purity measures how cell-type specific clusters are by comparing homogeneity of the external labels (FACS-defined cell types), within clusters provided by scRNA-seq data. AMI is a chance-adjusted information-theoretic measure of agreement between two labellings. This method accounts for artificially high mutual information between external labels and clusters when there is large number of clusters (see section 2.2 on page 38 for details on metrics). I also measured number of cells in robust clusters as quantitative metric. The robust clustering procedure allows cells to be omitted (not be assigned to a cluster) if they cannot be placed in a cluster across multiple clustering methods and/or parameters (see section 2.2 on page 38). MAGIC-processed data leads to a larger proportion of cells assigned to robust clusters, while *netSmooth* and scImpute lead to a reduction in the clustering robustness metric (Figure 2.3a). All three methods are able to discover some novel clusters in the data with high purity (Figure 2.3b). A closer inspection shows that MAGIC achieves this through a proliferation of small clusters, which are not, so far as I could judge, meaningful beyond chance. This is evidenced by the lower adjusted mutual information score (Figure 2.3c). *netSmooth*-preprocessed clusters achieve a higher AMI score, which, while modest, is biologically relevant.

netSmooth improves embryonic development expression patterns in scRNA-seq

Next, I demonstrate *netSmooth* on 269 isolated cells from mouse embryos at different stages of pre-implantation development between oocyte and blastocyst, as well as 5 liver

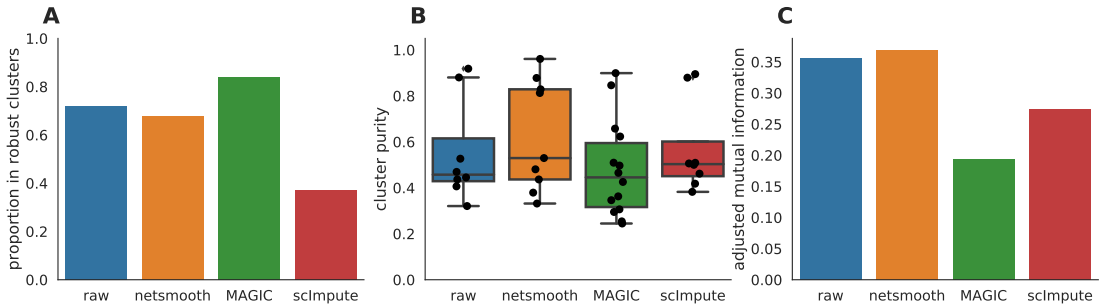


Figure 2.3: Hematopoiesis clustering metrics. **A)** The proportion of cells which were assigned to robust clusters. **B)** cluster purity (proportion of dominant cell type) for the robust clusters. netSmooth produces the most pure clusters in terms of cell types. **C)** AMI of the clustering results obtained after application of each of the methods. Only netSmooth increases the AMI between the clustering and the cell types. This figure is reproduced from Ronen and Akalin (2018a).

cells and 10 fibroblast cells sequenced and published by Deng et al. (2014). The authors of this study demonstrated that lower dimension embeddings capture much of the developmental trajectory (Figure 2.4a, Figure A.5a, A.6a). I used *netSmooth*, MAGIC, and scImpute on the scRNA-seq data to impute possible dropouts and reduce the noise. *netSmooth* and scImpute preserve most of the structure of the data, while MAGIC seems to push the data onto a completely different manifold (Figure 2.4 on page 54, Figure A.6 on page 118). I used the robust clustering procedure to obtain clusters, and computed the cluster purity and AMI metrics. *netSmooth* enabled the clustering procedure to place more of the samples into robust clusters (Figure 2.5a), and as in the hematopoiesis case, *netSmooth* is able to assist in identifying the developmental stage or tissue that cells belong to better than the other methods, as evidenced by the higher cluster purities (Figure 2.5b) and AMI (Figure 2.5c). scImpute also improves the cluter purity and AMI metrics (Figure 2.5b,c), and is not easily differentiable from *netSmooth* in the PCA scatter plot (Figure 2.4 on page 54). The *netSmooth* results are marginally better, which hints at an equivalence in the recovered signal quality between the two methods, *netSmooth*'s quasi-imputation incorporating priors, and scImpute's linear model-based imputation. scImpute achieves this by reducing the

overall 0-count genes significantly more than *netSmooth* (Figure A.7 on page 119), which suggests that incorporating priors the way *netSmooth* does can achieve similar results to data imputation, *without actually imputing too much*. The smaller change in the proportion of 0-count genes following *netSmooth* than scImpute (Figure A.7 on page 119) shows that *netSmooth* works primarily by smoothing values of genes with measured expression, as opposed to imputing suspected missing counts. This suggests a lesser transformation of the data, such as through application of *netSmooth*, can uncover much of the true signal hidden in the noisy data.

netSmooth aids identification of glioblastoma tumors

In a final analysis, I demonstrate applicability of *netSmooth* to cancer research. Patel et al. (2014) generated scRNA-seq data of 800 cells from 5 glioblastoma tumors and 2 cell lines. Lower dimension embedding plots show that cells from different tumors or cell lines generally group together, but some are not wholly distinguishable from other tumors (Figure 2.6a, A.8a, A.9a). Further, the two cell lines group closer to each other than the other patient samples. After applying *netSmooth* to the data, tumors become easier to distinguish in a lower dimensional embedding (Figure 2.6b), i.e. *netSmooth* improves assignment of each cell to its tumor, cell line, or clone of origin. Again, scImpute also leads to similar reduced dimension embedding (Figure 2.6d), while MAGIC distorted the data more than the other methods (Figure 2.6c). I used the robust clustering procedure before and after *netSmooth*, MAGIC, and scImpute. Only MAGIC increases the robustness of the clusters found in the data (Figure 2.7a), but *netSmooth* leads to the most pure clusters, in terms of tumor or cell line of origin (Figure 2.7b, Figure 2.7c).

Tumor or cell line of origin is an imperfect proxy for phenotypical variation in cancer

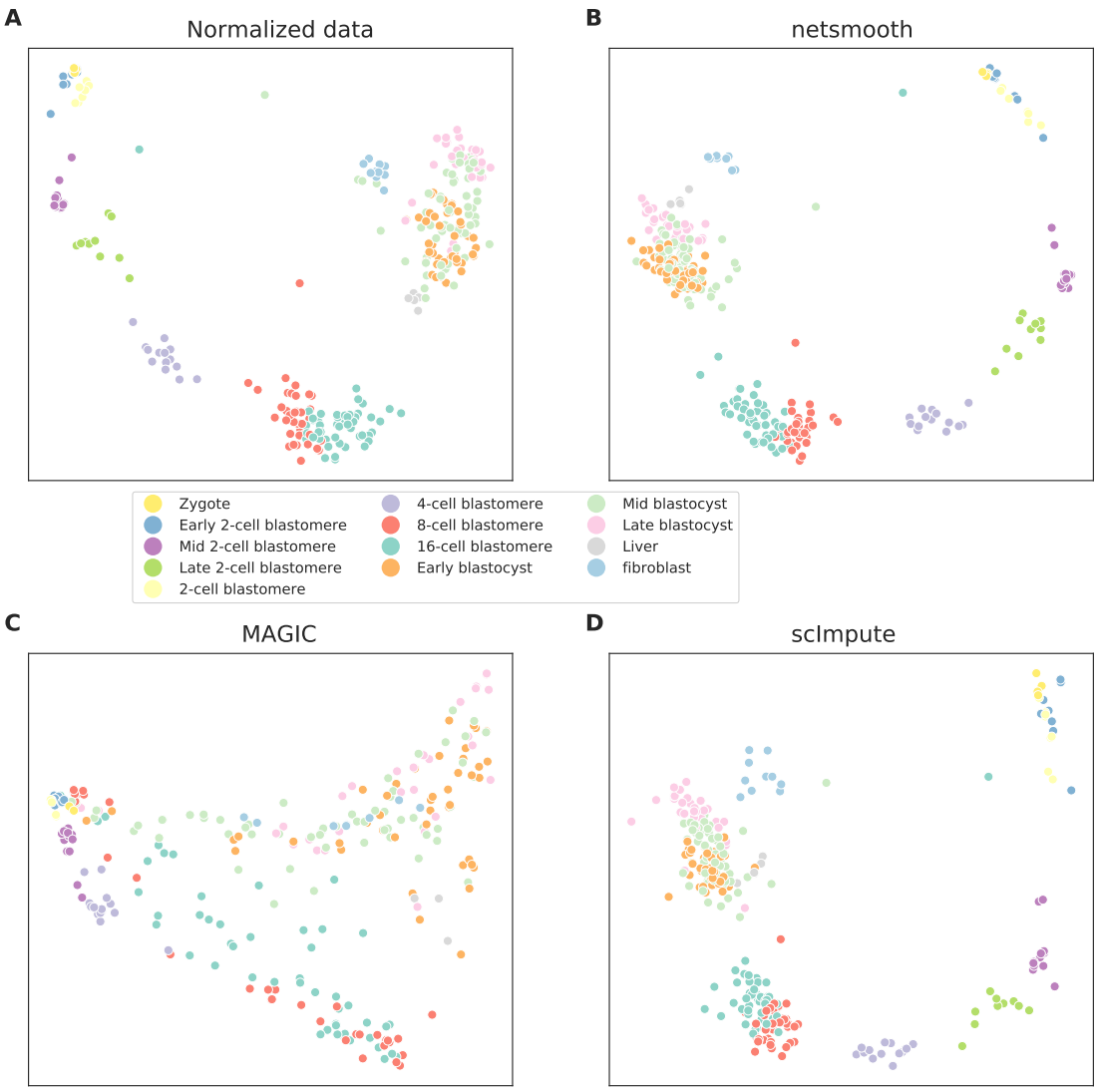


Figure 2.4: 2D PCA plots of the embryonic development dataset **A)** no preprocessing, **B)** after application of netSmooth, **C)** after imputing missing values with scImpute, and **D)** after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

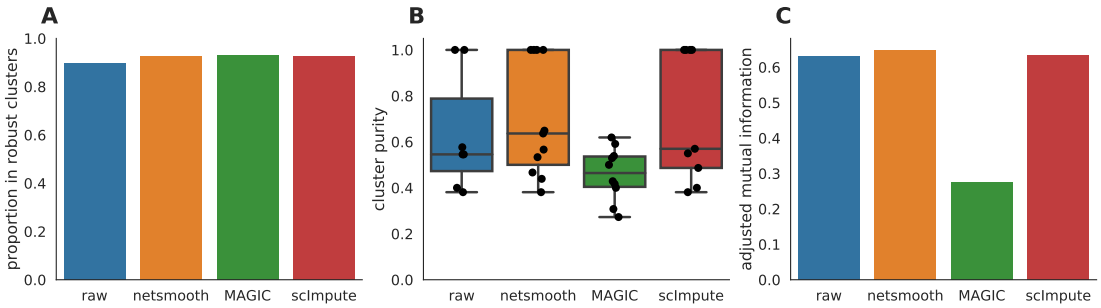


Figure 2.5: The Embryonic development dataset. **A)** The proportion of cells which were assigned to robust clusters. All three methods lead to better clusterability, with MAGIC having the strongest effect. **B)** cluster purity (proportion of dominant cell type) for the robust clusters. netSmooth produces the most pure clusters in terms of cell types. **C)** Adjusted mutual information of clusterings and cell types. Only netSmooth increases the AMI over the non-preprocessed data. This figure is reproduced from Ronen and Akalin (2018a).

cells, because some cells cluster by cell type rather than tumor of origin, demonstrating the heterogeneity in these glioblastoma tumors and similarities across origins (Patel et al., 2014). Nevertheless, I chose to compute cluster purity based on the cell origin rather than other labels which might be assigned to them, as it is the only *ground truth* variable that is independent of the RNA-seq experiment. Further, cells do group by origin (Figure 2.6 on the next page, Figure A.8 on page 120), and identification of origin is an interesting question in its own right in the field of cancer genomics, particularly for heterogeneous tumors such as these.

Sensitivity to the network

I SET OUT TO ENSURE that the results are not an artifact of the network structure, i.e. that the actual links between genes. We expect *netSmooth* not to perform well when using networks with similar characteristics, but where edges do not represent real interactions. To that effect, I constructed 20 random networks by keeping the same graph structure of the real PPI graph, but shuffling the gene names. Thus, these random networks share all the

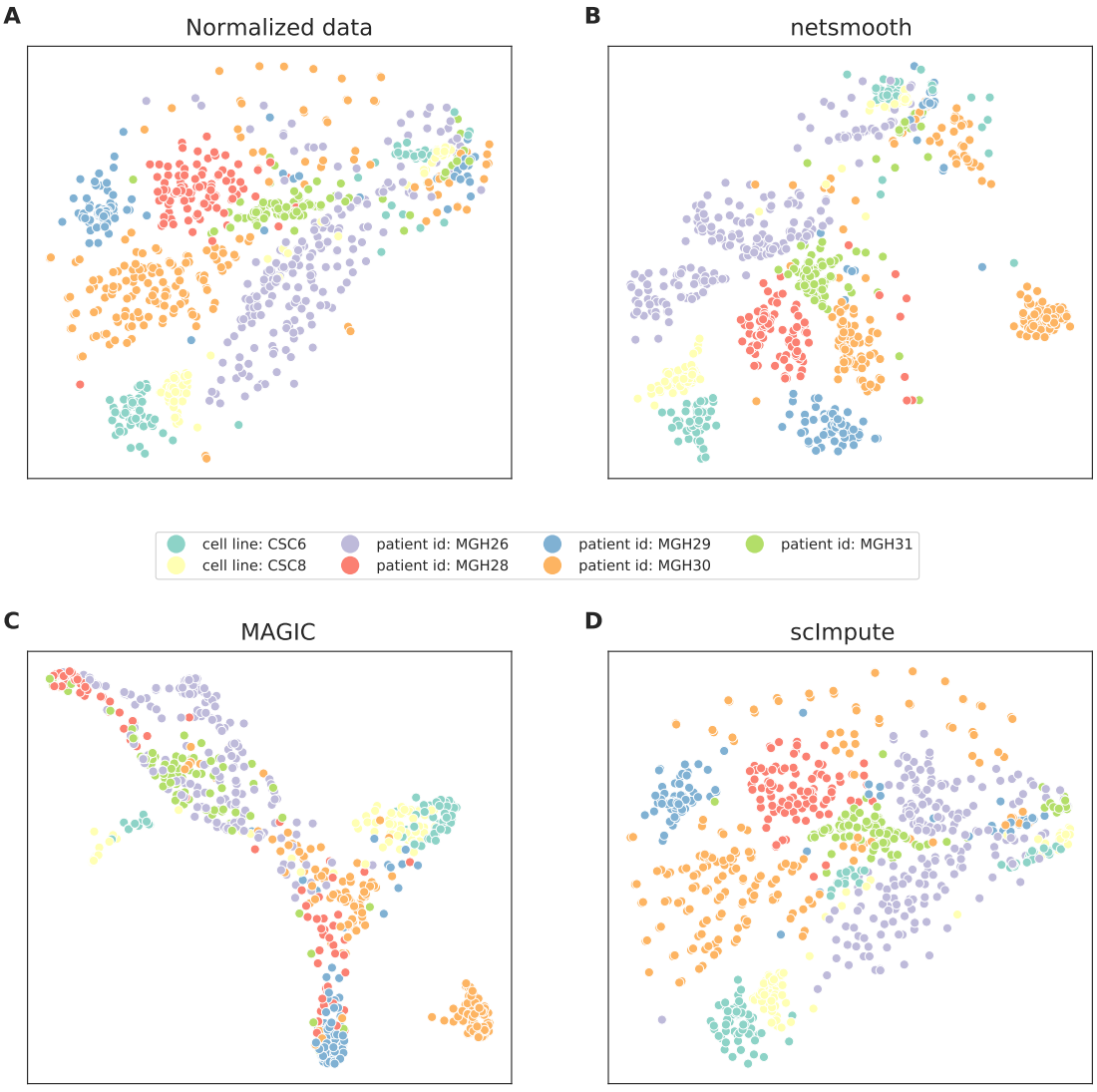


Figure 2.6: t-SNE plots of the glioblastoma dataset **A)** no preprocessing, **B)** after application of netSmooth, **C)**, using MAGIC, and **D)** after application of scImpute. This figure is reproduced from Ronen and Akalin (2018a).

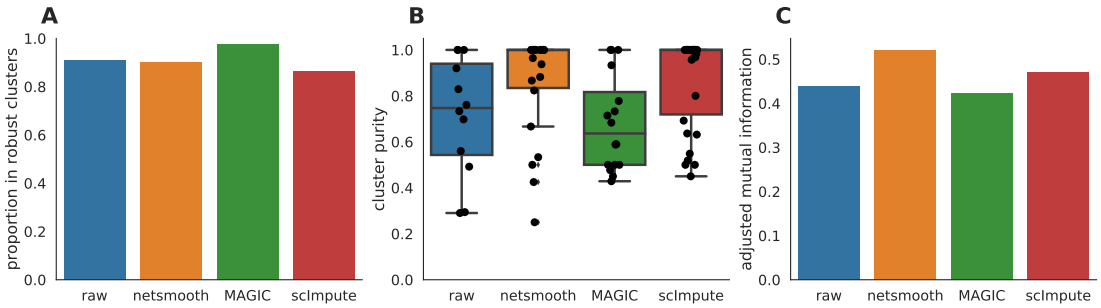


Figure 2.7: Imputation performance for the glioblastoma dataset. **A)** The proportion of cells which were assigned to robust clusters. netSmooth, MAGIC, and scImpute all increased the proportion of cells that are assigned to robust clusters, with MAGIC leading, netSmooth in second place, and scImpute in third. **B)** cluster purity (proportion of dominant cell type) for the robust clusters. netSmooth produces the most pure clusters in terms of tumor or cell line of origin. **C)** AMI of the clustering results obtained after application of each of the methods. This figure is reproduced from Ronen and Akalin (2018a).

characteristics of the real network (degree distribution, community structure, etc.), except for the true identity of the nodes. I then used those networks as inputs to *netSmooth* and ran the benchmarks as before on the hematopoiesis dataset. Using random networks as an input to *netSmooth* gives cluster purities distributed around a mode given by the cluster purities of the raw data, while the cluster purities given from using the real PPI network lie at the extreme edge of the distribution (Figure 2.8a). In other words, random networks are helpful or detrimental to cluster purity with approximately the same likelihood — and using the real gene network, we get an improvement as high as the best random networks can achieve. Further, most random networks result in fewer samples belonging to robust clusters (Figure 2.8b), that is, network-smoothing gene expression on random gene-gene networks tends to reduce the clusterability of the data, but much less so when using real gene-gene networks. Finally, I also calculated the adjusted mutual information of clusterings resulting from the randomized networks (Figure 2.8c). Again, most shuffled networks produce worse clusterings, with the real network outperforming all of them, as well as the no-smoothing case. I.e., network-smoothing gene expression on random gene networks tends to produce cluster results which are strictly worse than the raw data, while network-

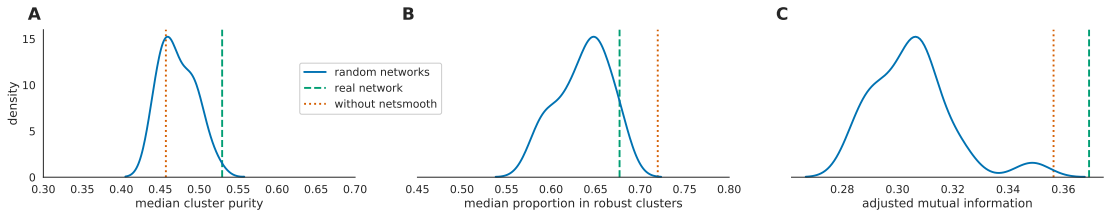


Figure 2.8: Performance of netSmooth with randomized networks. **A)** The median cluster purity achieved with the random networks. The real network outperforms the random ones, which result in cluster purities distributed around the purity given without using netSmooth. **B)** The proportion of samples assigned to robust clusters using the random networks as well as the real one. While all networks result in fewer samples robustly clustered (in the hematopoiesis dataset), the real network outperforms most random networks. **C)** The Adjusted Mutual Information achieved with the randomized networks. Most random networks produce clusterings with a worse AMI than using no network-smoothing. netSmooth with the real network structure produces the clustering result with the best AMI. This figure is reproduced from Ronen and Akalin (2018a).

smoothing on a real gene network improves the biological relevance of the clusters. This underlines the importance of also calculating the AMI, and not only the cluster purity, as cluster purity is sometimes improved using random networks (Figure 2.8a), but this is likely due to a proliferation of clusters some times. These results demonstrate that it is indeed the information contained in the PPI graph which enables *netSmooth* to transform the gene expression matrix in a more biologically coherent direction, and that the transformation we see can not be explained simply by the network topology, i.e. it only works if the edges in the network are real (network-adjacent genes really are co-regulated), and if so, it is worth doing.

Using other networks with netSmooth

In addition to using an unweighed (where all edge weights are 1), undirected (where all edge weights are positive) network from STRINGdb (see page 46), I constructed other gene networks and used them as inputs to *netSmooth*. I created a directed gene network

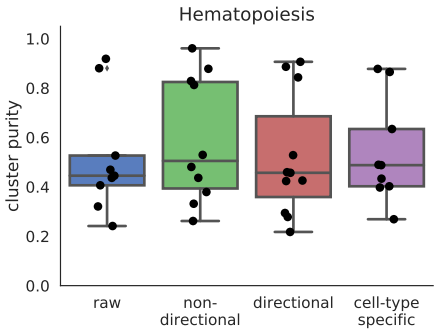


Figure 2.9: Cluster purities after applying netSmooth with different input networks. Raw refers to no smoothing, non-directional is the same as the results shown in previous sections. Directional refers to a gene network where inhibitory relationships have negative edge weights, and cell-type specific refers to a gene network of only genes which are known to have cell-type specific expression patterns. This figure is reproduced from Ronen and Akalin (2018a).

from only those edges in STRINGdb which are marked as activating or inhibiting*. I set the edge weights of the activating interactions to $+1$, and -1 for the inhibiting interactions, allowing gene expression values to be adjusted downwards for genes whose known antagonists are highly expressed. After smoothing, I set all negative smoothed expression values to 0. I also constructed a gene network from STRINGdb using only genes that are known to demonstrate cell-type specific expression. In order to obtain a list of genes with such cell-type specific expression patterns from the *Expression Atlas* (Petryszak et al., 2016), I used only the genes which show a cell-type specific expression with a mean transcripts per kilobase million (TPM) of at least 1 in some cell type, and used the subset of STRINGdb network containing those genes as an input to *netSmooth*. Both of those modified graphs perform similarly to the undirected graph from STRINGdb (Figure 2.9, Figure A.10a, Figure A.10b on page 122), demonstrating that *netSmooth* is able to use priors from different types of experiments in order to improve clustering of scRNA-seq.

I also considered other sources for the gene network. I constructed a gene network from HumanNet (Lee et al., 2011b), a functional gene network where edges denote interactions

*Most interactions in STRINGdb do not specify the direction, or nature of the interaction

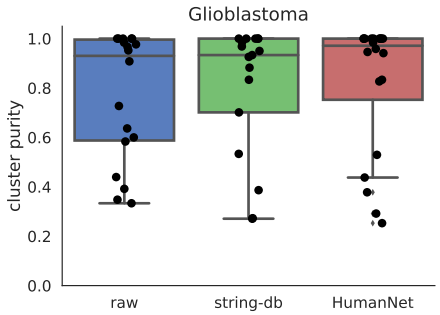


Figure 2.10: Cluster purities after applying netSmooth with different input networks. Raw refers to no smoothing, STRINGdb is the same as the results shown in previous sections, and HumanNet refers to a gene network constructed from the HumanNet database. This figure is reproduced from Ronen and Akalin (2018a).

between two genes. I constructed a smoothing graph by taking all edges from HumanNet, and producing a graph where all edge weights are set to 1. I then used this graph as an input to *netSmooth* on the glioblastoma dataset. It performs similarly to the network from STRINGdb (Figure 2.10, Figure A.10c on page 122), demonstrating that other sources for gene interactions may also be used by *netSmooth* to improve clustering results of scRNA-seq. Taken together with the results in the previous subsection, *netSmooth* improves scRNA-seq analysis, contingent on the priors (gene networks) being real.

As more scRNA-seq experiments are published, context-specific networks will be made possible to create, potentially improving *netSmooth*'s performance. The networks I have shown above have links between genes which are known in a general context, but scRNA-seq experiments might uncover previously unknown cell-type specific gene interactions, which could contribute to the information uncovered by network smoothing. Here, I have demonstrated that even the general-context networks I have used are able to assist in identifying specific cell types from noisy scRNA-seq datasets.

2.4 Discussion

Single cell RNA sequencing technology provides whole-genome transcriptional profiles at unprecedented throughput and resolution. However, high variance and dropout events that happen in all current scRNA-seq platforms complicate the interpretation of the data. Methods that treat 0 counts as missing values and impute them based on nonzero values in the data may amplify biases in the data.

I presented *netSmooth* as a preprocessing step for scRNA-seq experiments, overcoming these challenges by the use of prior information derived from protein-protein interactions or other molecular interaction networks. I demonstrated that *netSmooth* assists in several standard analyses that are common in scRNA-seq studies. This procedure enhances cell type identification in hematopoiesis; it elucidates time series data and assists identification of the developmental stage of single cells. Finally, it is also applicable in cancer, improving identification of tumor of origin for glioblastomas. In addition, I showed that the network smoothing parameter can be optimized by cluster robustness metrics, providing a data-driven way to pick the parameter when there are no other external labels to distinguish cells. I demonstrated that *netSmooth* can use prior information from different sources in order to achieve this.

I compared *netSmooth* with scImpute, a statistical genome-wide imputation method, and MAGIC, a genome-wide data smoothing algorithm, and demonstrated that while scImpute and MAGIC reduce the dropout phenomenon more than *netSmooth* does, *netSmooth* outperforms them in amplifying the biological/technical variability ratio. Thus, this is another voice added to the debate on the biological / technical origins of the dropout problem, and whether or not imputation is the best way to handle it. *netSmooth* provides clus-

ters that are more homogeneous and have higher adjusted mutual information (AMI) with respect to cell types. Although, in some cases data processed by MAGIC produces more robust clusters, the clusters returned after MAGIC processing do not have higher AMI or cluster purity. Higher robustness achieved by MAGIC processing might be due to the fact that the algorithm reinforces local structures too much in the data and produces artificially similar expression profiles between cells. Comparisons to CIDR (Figure A.11 on page 122, Figure A.16 on page 124) also show inferior performance to *netSmooth*.

In most of the benchmarks I ran, scImpute shows similar performance to *netSmooth*. The former relies on other data points in order to impute missing data, and the latter performs a quasi-imputation based on priors from other experiments. The analysis shows that *netSmooth* affects the dropout rate less than scImpute, while uncovering slightly more of the biological signal. This happens across the different overall dropout rates in the 3 experiments I profiled, indicating that *netSmooth* can achieve better results, with less obtrusive transformations of the data, than the imputation methods, across a range of experimental conditions.

Finally, *netSmooth* is a versatile algorithm that may be incorporated in any analysis pipeline for any experiment where the organism in question has a high quality gene network available. The algorithm is applicable to any omics data set that can be constructed as a genes-by-samples matrix, such as proteomics, SNPs, DNA methylation, and copy number variation. In Chapter 3, I will demonstrate another *netSmooth* use case, where I will show that network smoothing of mutations prior to multi-omics integration improves the survival prediction abilities of our models in cancer. In addition, most of the computational load of network smoothing can be done ”off-line”. As such it scales well with the number of cells, which is likely to increase in future scRNA-seq experiments. I have made available an R package to that end, which is available on GitHub: `https://github.com/BIMSBbioinfo/`

netSmooth, and Bioconductor: https://bioconductor.org/packages/release/bioc/html/netSmooth.html.

It was six men of Indostan

To learning much inclined,

Who went to see the Elephant

(Though all of them were blind),

That each by observation

Might satisfy his mind

John Godfrey Saxe, *The Blind Men and the Elephant*

3

Cancer sub-typing using priors from other experiments and deep learning multi-omics data integration

Analyzing data from a single ”omic” experiment, one would be wise to remember the parable of the blind men and the elephant; when each blind man grabs a different part of

the elephant, they will hardly agree on what an elephant is. In the same way, looking only at DNA mutations, or only at mRNA expression, two oncologists might not agree on what kind of tumor they are examining. Different omics assays may reveal different aspects of the same biological processes, sometimes in redundant ways, other times complementary to one another. Large scale initiatives such as The Cancer Genome Atlas (TCGA) have sequenced thousands of tumors using different omics protocols, empowering multi-faceted analysis of different cancer types. The TCGA data has enabled the field of cancer genomics to grow to maturity; since the first publication of the TCGA research network uncovered three previously unknown critical signaling pathways implicated in glioblastoma (Cancer Genome Atlas Research Network et al., 2008), and through the pan-cancer atlas (Hoadley et al., 2018) paper which revealed cancers arrising in different tissues can arrise from the same cells, the TCGA has made clear the clinical benefits that come from defining cancer sub-types using multi-omics data.

Artificial neural networks (ANN) have shown promise in many bioinformatics problem types, from sequence analysis through molecular structure prediction. In this chapter, I present a novel method I developed using ANNs as the backbone of a latent variable multi-omics integration framework. While the method is general, I demonstrate its utility in the study of colorectal cancer subtypes, and the assignment of colorectal cancer models (cancer cell lines) to sub-types. The following is adapted from Ronen et al. (2018), a paper on which I was the lead author. At the time of writing, the paper is undergoing peer-review, and the pre-print is available on bioRxiv.

Figure 3.1: Blind monks examining an elephant by Itcho Hanabusa. Ukiyo-e print illustration from Buddhist parable showing blind monks examining an elephant. Each man reaches a different conclusion based on which part of the elephant he has examined.

3.1 Introduction

Colorectal cancer (CRC) is a leading cause of death in the developed world. In the United States alone, it is predicted to cause over 50,000 deaths in 2019, second only to lung cancer (American Cancer Society, 2019). With over 140,000 new cases in the US yearly, this represents a five year survival rate of approximately 65%. While the mortality trend is decreasing for patients diagnosed at an age of 56 and up, most likely due to improvements in screening, patients diagnosed before the age of 55 face worsening prospects than before, underlining the poor treatment options available to CRC patients, with surgery being the most common treatment. This represents a pressing need for better understanding of CRC sub-types, as patient populations with distinct molecular disease sub-types are likely to benefit from drug discovery trials targeting their disease.

The CRC label encompasses different diseases with distinct morphological and molecular characteristics. Common molecular aberrations include mutations in the WNT, MAPK, TGF-β, and PI3K–AKT signaling pathways (Parsons et al., 2005; Fearon, 2011; The Cancer Genome Atlas Network, 2012). These mutations occurr through different mechanisms: chromosomal instability (CIN), which is characterized by many copy number alterations (CNA), and microsatellite instability (MSI), leading to hyper-mutated tumors, but with mostly diploid genomes (low CNA) (Müller et al., 2016). These mechanisms may be driven by mutations in e.g. DNA mismatch repair (MMR), or through epigenetic alterations (Lao and Grady, 2011). Müller et al. (2016) analyzed the colorectal cancer TCGA cohort and classified them into hyper-mutated ($\approx$ 16%) and non hyper-mutated ($\approx$ 84%) cancers. The hyper-mutated cancers have MSI. The non-hypermutated, microsatellite stable (MSS) cancers, are characterized by CIN, with high occurance of CNA, and mutations in the APC, TP53, KRAS and BRAF genes. Some of the tumors also have the CpG island methylator phenotype (CIMP, Toyota et al. (1999)), which epigenetically silences tumor suppressors and MMR genes, confirming that the TCGA data was representative of current understanding of CRC.

The gold standard in molecular sub-typing of CRC is the Consensus Molecular Subtypes (CMS), developed by the colorectal cancer subtyping consortium (Guinney et al., 2015). This classification system, based on gene expression profiling of thousands of CRC tumors, divided tumors into four subtypes with distinguishing features. The CMS1 subtype is defined by hypermutation, microsatellite instability and strong immune activation; CMS2 is defined by chromosomal instability (CIN), WNT and MYC signaling activation; CMS3 is defined by metabolic dysregulation; and CMS4 is defined by growth factor β activation, stromal invasion, and angiogenesis. Approximately 13% of the tumors do not belong to a consensus subtype, as they have mixed gene expression signatures. They may

represent distinct tumor types, or samples with intra-tumor heterogeneity. While the CMS classification is based on gene expression, follow-up analysis of the tumors revealed distinct copy number profiles, mutation frequencies, and methylation profiles (Guinney et al., 2015) (Figure B.6 on page 130), indicating that other omics types contain important information.

I developed a multi-omics integration method that can incorporate gene expression, copy number, and mutation data to identify CRC subtypes, with implications for patient stratification. I used it to refine the CMS classification system for CRC tumors, and to assign colon cancer cell lines to the different sub-types. As xenografts and 3D organoid tumor models become more widespread, the method will be able to match those to sub-types as well. This will be highly useful, as these have been shown to predict drug response in patients (Vlachogiannis et al., 2018). By leveraging multi-omics datasets, CRC samples that can not be associated to a CMS subtype are also assigned to an appropriate subtype. In addition, multi-omics signatures, incorporating gene expression, point mutations and copy number alterations, are a direct output of the method, and will not need to be generated post-hoc, e.g. by examining mutation rates in groups defined by gene expression profiles, as in the CMS.

The method I developed uses deep learning to perform latent factor analysis, an *unsupervised learning* technique. Genomic assays are high-dimensional (tens of thousands of genes), and high-dimensional spaces are challenging to analyze. This problem is further exacerbated by the introduction of multiple assays from different omics platforms. Latent factor analysis seeks to find a lower dimension representation of the data, which preserves the important structure and patterns therein. This is sometimes referred to as dimensionality reduction. By describing the data using a handful of *latent factors*, rather than tens of thousands of genes, down-stream analysis, such as distance calculations and clustering, are simplified (Trunk, 1979). It is further desirable that latent factors be interpretable in the biolog-

ical context, i.e. that the patterns summarized by a latent factor implicate cellular processes. The workhorse for latent factor analysis for multi-omics, as well as general data analysis, has been matrix factorization (Tini et al., 2017). Latent factor analysis for multi-omics data typically includes concatenating the different omics data to a single matrix and applying a well-known matrix factorization algorithm, sometimes with weighting of the individual data sets. Multifactor analysis (MFA, de Tayrac et al. (2009)) and iCluster+ (Mo et al., 2013) are examples of such methods. Some such algorithms, such as MFA, impose orthogonality of factors, i.e. that the factors explain disjoint underlying processes, as in PCA. Orthogonality might be conceptually appealing, but is not a biological necessity. Orthogonal latent factors may be the best for statistical reconstruction of a dataset, and still be biologically difficult to interpret. One feature of latent factor representations which aids biological interpretability is sparsity, i.e. each latent factor only depends on a few of the input genes, and each sample is described by only a handful of latent factors. Sparsity in the relationship between input genes and latent factors simplifies the task of biological interpretation of the model, that is, implicating known biological processes underlying the latent factors; sparsity in the relationship between latent factors and samples simplifies down-stream analysis such as clustering.

While it is reasonable to expect that the latent factors describing the data be of much lower dimensionality than the genome-scale input data, it is an open question just how low the dimensionality should be. Heuristics to pick the number of latent factors have been proposed by method designers. PCA and MFA typically suggest an *elbow* method, where latent factors are ordered by their variance explained, and the user determines an *elbow* point in the graph, discarding latent factors with a low variance explained. MOFA formalizes this heuristic by starting the fitting process with a high number of latent factors, and during training, discarding ones with a variance explained below a pre-set threshold (with

a default value of 2%). iCluster+'s heuristic comes from its k-means roots, i.e. one tends to set the number of latent factors to $K - 1$ where K is the number of clusters one expects to find. Recently, a large-scale study of latent factor methods (Buhai et al., 2019) has demonstrated the desirability of specifying more latent factors than are expected to exist in the data. Specifically, it was shown that using more latent factors than are needed, can compensate for shortcomings in the training algorithm, and improve log-likelihood and recovery of ground-truth latent factors. Hence, it is desirable to have a latent factor method that is able to learn a large number of latent factors efficiently from the data.

I developed my method building on disentangled variational autoencoders (β-VAE) (Higgins et al., 2017), an unsupervised deep learning architecture which has been shown to generalize well and produce disentangled representations. A similar method has been shown to be able to stratify cancers by their tissue type based on gene expression profiles (Way and Greene, 2017), and other autoencoder architectures have been used to integrate multi-modal data in robotics (Cadena, Dick, and Reid, Cadena et al.), as well as protein function prediction (Gligorijević et al., 2018) and small molecule characterization (Winter et al., 2019). I use a multi-modal, stacked β-VAE to extract latent factors which are important for defining subtypes and predicting patient survival. I call the method, "**M**ulti-omics **A**utoencoder **I**ntegration", or, *maui*.

In this chapter, I will describe *maui* and compare its performance to other state-of-the-art multi-omics data integration methods. I will use it to refine the CMS classification system, and to match cancer cell lines to the different sub-types. I will also demonstrate how combining the two methods I developed, *netSmooth* (Chapter 2) and *maui*, results in even better survival prediction for CRC patients.

I made *maui* available as a Python package, so that other researchers may use it with

their own data. At the time of writing, it has been downloaded more than 6,000* times. It is easily obtainable from GitHub (http://github.com/bimsbbioinfo/maui), and PyPI (https://pypi.org/project/maui-tools/).

3.2 Methods and Data

Data

I DOWNLOADED DATA FOR TUMORS from the TCGA-COAD (n=389) and TCGA-READ (n=130) project designations of the Genomic Data Commons † using the *TCGAbiolinks* R package (Colaprico et al., 2016). I downloaded the CMS annotations for the TCGA tumors from the Colorectal Cancer Subtyping Consortium (CRCSC) ‡. Table 3.1 summarizes the subtype information. The gene expression data (mRNA) is HTSeq - FPKM. Mutations were downloaded as MAF files, filtered to include non-synonymous mutations only, and represented as a binary mutation matrix where $m_{ij} = 1$ if and only if gene i carries a non-synonymous mutation in sample j. Copy number alterations are GISTIC (Beroukhim et al., 2007) calls by gene, represented as a real-valued matrix where c_{ij} is the GISTIC segment mean for the segment containing gene i in sample j. Cancer Cell Line Encyclopedia data was obtained from the CCLE portal §, and is the same data types as the TCGA data, with the exception that transcriptome profiles are RPKM-normalized and not FPKM. I considered 54 cancer cell lines originating from the colon.

I considered only tumors (from TCGA) and cancer cell lines (CCLE) which have "complete data", that is, available measurements in all three assays: gene expression, SNVs, and

*https://pepy.tech/project/maui-tools
†https://portal.gdc.cancer.gov
‡http://sagebionetworks.org/research-projects/colorectal-cancer-subtyping-consortium-crcsc/
§https://portals.broadinstitute.org/ccle

CNVs.

CMS label	Description	# Samples
CMS1	MSI Immune	61
CMS2	Canonical	175
CMS3	Metabolic	60
CMS4	Mesenchymal	123
Total with CMS label		**419**
Without CMS label		100
Total		**519**

Table 3.1: Summary of TCGA tumors' CMS labels. This table is reproduced from Ronen et al. (2018).

I used gene-wise MAD statistic, computed directly on the raw data described above, in order to select the most informative genes. For the comparisons with MOFA and iCluster+, I used the 1,000 genes with the highest MAD for gene expression, 200 for mutations, and 100 for copy number alterations, for a total of 1,300 input features. I selected the features so strictly in order to make a comparison against iCluster+ viable, and with a larger feature space iCluster+'s runtime would become untenable (Table 3.2 on page 91).

For the final clustering analysis, I used a larger feature set, with 5,000 gene expression values, 500 mutations and 500 CNVs for a total of 6,000 features, taking advantage of *maui*'s neural network architecture which allows for larger feature spaces to undergo feature selection as part of the training.

I fit maui using all TCGA samples, both with and without a CMS label (n=519, Table 3.1) as well as colon-derived cancer cell lines (n=54), for a total training set size of 573. For the analysis that depends on a CMS label being available, the input features were the latent factors, and the samples only those TCGA samples with a well-defined CMS label (n=419, See Table 3.1).

All input features were scaled and centered prior to feeding to the neural network, using batch normalization. Prior to this scaling, mutation data was binary, CNV data GISTIC

calls, and gene expression counts were RPKM/FPKM values (Mortazavi et al., 2008) which were scaled and centered. TCGA and CCLE gene expression matrices were first scaled and centered individually, and then concatenated and scaled jointly, in order to filter out the "batch effect" of CCLE vs. TCGA data and enable mapping of tumors and cell lines to the same space. This means that, for a trained *maui* model, when new, unseen samples are to be mapped onto the latent factor space, they must first be normalized in this way to fit the distribution of the training data.

Latent factor model for multi-omics data

Starting from different data matrices x_i from different modalities, we call the full multi-omics data set $x = [x_1, x_2, ..., x_m]$.

We define a generative model $x \sim p(x|z)$. Graphically, our model looks like Figure 3.2a, a Bayesian latent variable model where the variation in the data x is explained by the variation in a smaller set of latent factors, z. In order to infer the latent variables $z \sim p(z|x)$, as $p(z|x)$ is generally intractable, we proceed with a variational Bayes framework, i.e. approximating $p(z|x) \approx q_\theta(z|x)$, where $q_\theta(z|x)$ is a simple class of distribution, and minimizing the Kullback-Leibler divergence $D_{KL}(q_\theta(z|x)||p(z|x))$. This is equivalent to maximizing the Evidence Lower Bound (ELBO) (Blei et al., 2016):

$$ELBO = E_q[log(p_\phi(x|z))] - D_{KL}(q_\theta(z|x)||p_\phi(z)).$$

We follow Kingma and Welling (2013) and re-parameterize z_i^l as

$$z_i^l = \mu_i + \sigma_i \epsilon_l$$

where

$$\epsilon_l \sim \mathcal{N}(0, \mathbf{I}),$$

which allows us to construct the Autoencoder shown in Figure 3.2b.

The first half of the autoencoder, leading from x to z (the "encoder") is a neural network which will be trained to compute $q_\theta(z|x)$, that is, θ denotes the weights of the encoder network. The second half, the "decoder" network, is a neural network which will be trained to compute $p_\phi(x|z)$, so ϕ denotes the weights of the decoder network. Thanks to the reparametrization of z, the path from x to $\hat{x}$ is differentiable, via backpropagation, in θ and ϕ, and thus we can use gradient descent to optimize a loss function that is differentiable in θ and ϕ.

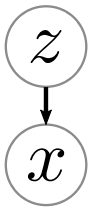
Setting the loss function of the neural network to the negative ELBO

$$l = -E_q[log(p_\phi(x|z))] + D_{KL}(q_\theta(z|x)\|p_\phi(z)),$$

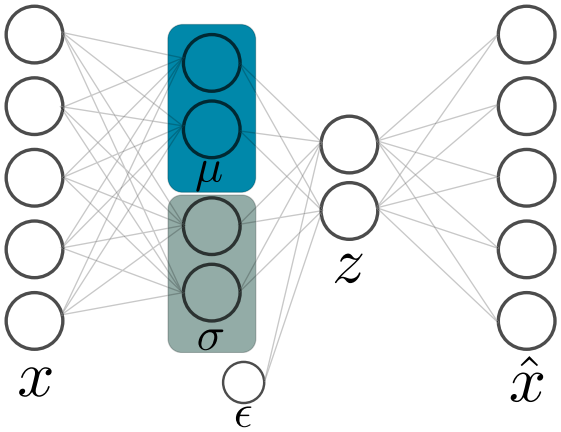
we see that the first term is equivalent to the cross-entropy reconstruction loss, and the second term, the KL-divergence between $q_\theta(z|x)$ and the prior $p_\phi(z)$ can be seen as a regularization term, which will push the z's to their prior distribution.

Stacking autoencoders

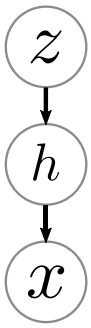
The Variational Autoencoder described above is for a one-layer bayesian framework, i.e. Figure 3.2a. But Autoencoders may be stacked (Bengio et al., 2007) to produce deeper neural network architectures. Deep architectures have more than one layer of nonlinearities, and can thus more compactly capture highly nonlinear functions. We introduce a hidden layer to our Bayesian latent variable model (Figure 3.2c).



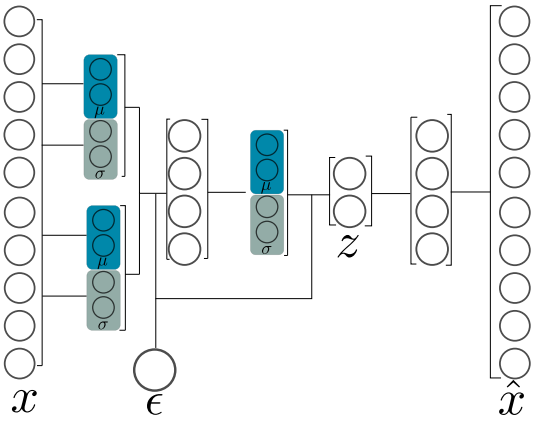
(a) Plate model of latent factor model; $x \sim p(x|z)$.



(b) Visual representation of Variational Autoencoder.



(c) Plate model of stacked latent factor model.



(d) Visual representation of Stacked Variational Autoencoder.

Figure 3.2: Graphical models and neural network schematics of corresponding Autoencoders. a, b: latent variable model. c, d: multilevel latent variable model. This figure is reproduced from Ronen et al. (2018).

Using the reparametrization trick as above, and specifying the full loss function, inference in the generative model (Figure 3.2c can be done by backpropagation in the stacked variational autoencoder model (Figure 3.2d).

Model regularization

Deep neural networks have many parameters, making them very flexible. This flexibility, however, comes at a cost—deep models are prone to over-fitting: the generation of models which explain the training data well, but generalize poorly to new data. In addition, deep nets are prone to producing complex relationships between many variables. In the case of a latent variable model, that means latent factors that change with the variation of any of a large number of input features, a property which makes the task of interpreting the biological meaning of those latent factors difficult. In technical terms, we wish to enforce sparsity in $q_\theta(z|x)$, so that each latent factor will depend on fewer of the inputs.

In order to address the first issue of potential over-fitting, I use Batch Normalization (Ioffe and Szegedy, 2015). When fitting the model, we segment the data into mini-batches, at each iteration computing derivatives and making updates to the model based on that sample. Using Batch Normalization, each feature is scaled and centered in each mini-batch. We feed all of the training examples to the model fitting procedure until the entire training set is exhausted, and then we segment it into new minibatches and repeat the process, for a specified number of epochs. This way, each time a training sample is passed to the model, it will be slightly different, which is roughly equivalent to adding noise, which has been shown to work as a regularizer in Denoising Autoencoders (Vincent et al., 2010) and prevent over-fitting. Further, Batch Normalization addresses another issue - that of Internal Covariate Shift. Internal Covariate Shift happens when the distributions of activations of

internal nodes in the neural network changes while training. Reducing Internal Covariate Shift enables us to pick higher learning rates, and thus speeds up inference considerably.

The second mode of regularization, encouraging representations where latent factors depend only on a few input features, is partially achieved by the KL term in the loss function, as that penalizes distributions of z's which are far from the Gaussian prior. Disentangled representations, where latent factors depend on complementary input feature sets, can support this kind of sparsity. This holds when latent factor representations are non-negative, which we achieve by passing them through a rectifier unit (ReLU). When each non-negative latent factor depends on a different set of inputs, the relationships will be sparser.

We therefore add a multiplier to the loss function similar to β-VAE Higgins et al. (2017) and , allowing us to weigh the relative importance of the terms:

$$l = -E_q[log(p_\phi(x|z))] + \beta D_{KL}(q_\theta(z|x)||p_\phi(z))$$

In order to ensure the network finds a good representation before it starts regularizing, we use the "warm-up" method proposed by Sønderby et al. (2016), where β is initially 0, and is gradually increased by $\beta = \beta + \kappa$ until its value reaches 1.

Model implementation

The model was implemented using Keras (v2.1.5) with a Tensorflow (v1.6.0) backend. I used Rectified Linear Units for all activations except for the last layer which is Sigmoids, for all features. I trained our network for 600 epochs using mini-batches of size 100 and $\kappa = .01$. I used the Adam optimizer (Kingma and Ba, 2014).

Predicting CMS from latent factors using SVM

In order to quantify the correspondance between latent factors learned by different methods, adn the CMS label, I used Support Vector Machines (Hastie et al., 2001) (SVM), a supervised learning algorithm. I used a regularized SVM, picking the optimal regularization parameter using 10-fold cross-validation (CV). Then, I predict the CMS label of each sample out-of-sample, when it is in the test set, using 10-fold CV. Receiver operating characteristic (ROC) curves were computed for each class by modeling a binary outcome for each CMS label (one-vs-all). Mean ROC curves were computed by averaging the ROC of all CMS labels at each point.

Unsupervised prediction of CMS using k-means clustering

I benchmarked *maui* against MOFA and iCluster+ in the power of latent factors to predict CMS labels in an unsupervised fashion, in order to present a fair comparison between *maui* (70 latent factors) and MOFA (20 latent factors) and iCluster+ (10 latent factors). I used k-means clustering, as clustering based on distance metrics suffers from the "curse of dimensionality", and does not, in general, benefit from a larger number of input dimensions (unlike supervised learning methods). To assess the ability of k-means clusters to capture the CMS labels, I ran k-means with 1,000 starts, picking the best (lowest variance) solution for each run. In addition, I applied the algorithm with K's in the range of 2–9. For the cluster assignments for each K, I computed the Adjusted Mutual Information (AMI) of the clustering with the CMS labels. The AMI is an information-theoretic measure of the concordance between two labelings (clusterings and CMS), which accounts for chance.

Higher values indicate closer relationships between labelings.

A novel subtyping scheme for CRC with cell line associations to subtypes

The subtyping scheme presented in the results section is based on k-means clustering using *maui* latent factors learned from multi-omics data. I did this using a *maui* model trained on 6,000 input features (5,000 gene expression, 500 mutation, 500 CNV), as it is more predictive of patient survival than the one using 1,300 features (Figure 3.3F), and produces largely the same cluster assignments as the 1,300 gene model presented above (Figure B.12).

Association of latent factors with genomic features

The Stacked Variational Autoencoder model described above computes latent factors $z = f(x)$ where $f(x)$ is a nonlinear function which may not necessarily be well-approximated by a linear $z \approx Wx$, as in models such as MOFA or iCluster+. The architecture and depth of the neural network also makes it nontrivial to associate the input genomic features (gene expression, mutations, etc.) with the different latent factors. However, in order to make biological sense of the latent factors, it is necessary to make that association. In order to do that, I computed Spearman's ρ for each latent factor with each input feature, and call a latent factor associated with an input feature if $p < 0.001$.

Gene set enrichment

In order to identify genes associated with the different clusters, I performed a differential expression analysis using t-tests and Benjamini-Hochberg correction for multiple hypothesis testing. Genes with adjusted p value below 0.05 were called differentially expressed. In order to find out if the genes associated with latent factors (Figure 3.5), or with clusters (Figure 3.4) belong to known pathways, I used *Enrichr* (Chen et al., 2013; Kuleshov et al., 2016) via the python package *gseapy* [*]. I used pathways (gene sets) defined by KEGG (Kanehisa, 2000; Kanehisa et al., 2015, 2016).

Survival analysis

I rely on overall survival data from the TCGA annotations for all survival analyses.

In order to assess the prognostic value of latent factors inferred by our deep learning approach, I fit a Cox Proportional Hazards model (Breslow, 1975),

$$ln\frac{h(t)}{h_o(t)} = \sum_i \beta_i x_i,$$

where the left hand side is the logarithm of the hazard ratio, and x's are co-variates. I assess the predictive value of each latent factor separately, while controlling for the patient's age, gender, and tumor stage at diagnosis. I compute confidence intervals for the coefficient β associated with the latent factor, and pick the latent factors with FDR-correction and $\alpha = 0.95$.

[*] version 0.9.4, available from PyPI https://pypi.org/project/gseapy

In order to compare the prognostic value of different models, we compute Harrel's C index (Harrell et al., 1982, 1984, 1996) and use 5-fold cross-validation (Hastie et al., 2001).

The log-rank statistics reported in Figure 3.3D and Figure B.3 are multivariate log-rank test, under a null hypothesis that all groups have the same survival function, with an alternative hypothesis that *at least one group* has a different survival function.

All survival analysis was done using the python package *lifelines* [*].

Comparing models' survival-predictive value

In order to compare *maui* to MOFA and iCluster+ (as well as to a gene expression only-based *maui* model), I used Harrell's C in a Cox Proportional Hazards regression model (Breslow, 1975; Pencina and D'Agostino, 2004). The c-Index was computed for Cox models based only on clinically relevant factors, which I select using individual, unregularized Cox models, one per factor, while controlling for patient age, sex, and tumor stage. Using unregularized models ensures this first feature selection is done in an unbiased fashion. In those individual factor models, I used Efron's method to compmute confidence intervals, and only kep the latent factors with statistically significant (adjusted P-value < 0.05) nonzero coefficients in the individual Cox models. Having selected clinically relevant latent factors from each model (*maui*, MOFA, iCluster+, *maui*—expression, *maui*—*netSmooth*), we fit a full Cox regression using those, and ran a cross validated out-of-sample c-Index calculation using regularized Cox PH regression, searching for the optimal result among the regularizers $1, 10, 100, 1000, 10000$. The results reported in Figure 3.3F are the best regularized model for each of the methods.

[*] https://lifelines.readthedocs.io/en/latest/

Model selection

The stacked VAE presented above is a class of models which are parameterized by the number of hidden units (the dimensionality of h), N_{hidden}, and the number of latent factors, N_{latent}.

This presents an opportunity for selecting the best model by spanning a grid over the two parameters, and computing some score. We searched the space spanned by (N_{hidden}, N_{latent}) and computed a compound benchmark score at each point. The compound benchmark score is the average of the scores of: the AUC in the supervised CMS prediction task, the AMI in the unsupervised CMS subtype prediction task, the $-log_{10}p$ of the multivariate log-rank test for differential survival statistics, and the c-index Pencina and D'Agostino (2004) from the Cox proportional hazards model. Maui is largely insensitive to the choice of (N_{hidden}, N_{latent}), for $N_latent > 30$ (Figure B.10).

MOFA was run using the default parameters. It uses heuristics in order to pick the number of latent variables, starting with a large number and pruning away ones with explained variance ratio of beneath a threshold of 2%. The resulting model had 20 components. In order to see if MOFA's heuristic picks a sensible model, we also ran it with fixed numbers of latent factors over a range from 10 to 30, and computed the composite benchmark, lower than for maui due to the higher runtime.

For iCluster+, there are two free parameters: the regularization parameter λ, and the number of latent factors. We ran a grid search over the regularization parameter and number of latent variables space, similar to the way maui was tuned, but with a lower number of maximum latent factors, due to iCluster+'s prohibitive runtime for larger numbers. For each parameter configuration, we computed the compound benchmark. maui consistently

outperforms both MOFA and iCluster+ for most parameter sets (Figure B.11).

For the final analyses shown in the results section, in order to avoid leakage of benchmarks into the unsupervised learning algorithms, we ran maui and with parameters corresponding the the mean of the distribution of compound benchmarks ($N_{hidden} = 1100$ and $N_{latent} = 100$); the same reasoning for iCluster+ resulted in 5 latent factors. We allowed MOFA to use its own heuristic, discarding latent factors with variance explained below 2%, yielding a 20 component model. We used the MOFA default threshold when picking the number of components to keep in the PCA comparison, which yielded 5 components.

Quality assessment of CRC cell lines for modeling tumors

In order to assess the fitness of different cancer cell lines as models for tumors, I computed the pairwise euclidean distance between each of the samples (TCGA and CCLE), in the space of the latent factors derived from *maui*. Then, I computed, for each cell line, the proportion of its 5 nearest neighbors which are also cell lines, the working hypothesis being that cell lines that form "cell line clusters" are more cell-line like than tumor like, and likely less fit as models for tumors. I repeated the exercise considering other numbers of nearest neighbors from 1—20, at each K computing the true positive rate (recall), that is, $\frac{\text{\# non-colon cell lines predicted to be poor models}}{\text{\# of non-colon cell lines}}$, showing that the recall is near perfect for a wide range of K's.

3.3 Results

In the pages that follow, I will use *maui* to find a latent factor representation of CRC tumors and cancer cell lines from multi-omics data. I will use this latent factor rep-

resentation to define sub-types of the disease, refining the CMS classification scheme. I will demonstrate that these sub-types are bioligically distinct, and carry clinical implications. I will show that *maui* performs better than state-of-the-art methods for multi-omics integration. Finally, I will demonstrate how *maui* can be used for quality control of cancer models, and to assign them to cancer sub-types.

Refining CRC subtypes using multi-omics data

The CRC cohort in the TCGA data-set (n=519, See Materials and Methods) has been extensively studied, and a state-of-the-art subtyping scheme exists in the "Consensus Molecular Subtypes" for colorectal cancer, or CMS (Guinney et al., 2015) (Table 3.1). In order to validate that the latent factors learned by *maui* capture patterns relevant to cancer biology, I tested how well the latent factors recapitulate the known subtypes.

I extracted latent factors from gene expression, point mutations, and copy number alterations using *maui*, as well as other published methods for multi-omics integration by dimensionality reduction: MOFA (Argelaguet et al., 2018), and iCluster+ (Mo et al., 2013). In order to quantify the relationship between latent factor representations and the CMS subtype to compare the methods, I used Support Vector Machines (SVM, See Methods) to assign a CMS label to each tumor based on their latent factor representation. I then computed Receiver operating characteristics (ROC), and computed the area under the curve (see Methods). The area under the ROC (auROC) is a measure of classification accuracy, with a score of 0.5 being expected from random guessing, and 1.0 being perfect. All methods produce latent factors with some correlation to the CMS labels (Figure 3.3A). Using SVM, *maui* (auROC 0.98) marginally outperforms MOFA (auROC 0.96) and both *maui* and MOFA dramatically out-perform iCluster+ (auROC 0.73) (Figure 3.3B, Figure B.2).

maui has an unfair advantage over MOFA in the previous analysis, as I ran it with 80 latent factors, whereas MOFA was only run with 20, and regularized supervised learning algorithms may benefit from a larger number of input features (here, the latent factors). In order to assess which of the methods is best able to capture the CMS labels, without regard to the number of latent factors, I repeated the previous exercise—predicting the CMS from the latent factors—using an *unsupervised learning* algorithm. I clustered the samples with a well-defined CMS [*] using k-means clustering on the latent factors (See Methods). I let K vary from 2—9, and for each K, computed the Adjusted Mutual Information (AMI) of the clustering with the CMS labels. k-means clustering only reproduces the CMS subtype to a significant degree for K values of 4—6, and only using *maui* (Figure 3.3C). This clustering analysis shows that *maui* factors are superior at predicting CMS labels, in a fair comparison, as k-means clustering is based on distances, whose computation does not benefit from higher dimensionality — in fact, the opposite is true (Trunk, 1979).

Latent factors inferred by *maui* are predictive, using k-means, of the CMS subtype, especially using K's 4—6 (Figure 3.3C). In order to pick the best clustering result to focus on, I computed the log-rank statistic for significance of differential survival rates between clusters (See Methods). $K = 6$ results in the most statistically significant survival difference ($P < 0.001$, Figure 3.3D). Note that the CMS subtypes on their own are not indicative of survival rates in the TCGA data ($P = 0.77$), and that maui with $K = 4$ ($P < 0.045$) and $K = 5$ ($P < 0.019$), also produce clusters with significant differential survival (Figure B.3). Notably, $K = 6$ is preferable to $K = 4$ and $K = 5$, as it is able to tease out a cluster with particularly poor prognosis (cluster 3), which consists mostly of a subset of tumors with the CMS2 (Canonical) designation (Figure B.3). K-means clustering of MOFA or iCluster+ latent factors does not produce statistically significantly separable clusters (Figure B.4,

[*] Some CRC samples do not have a consensus molecular subtype

Figure B.5).

I also compared the ability of *maui*, MOFA, and iCluster+ to predict patient survival, irrespective of any clustering. I first selected, for each model, a subset of latent factors which are individually predictive of patient survival, and call those *clinically relevant* latent factors (See Methods). Using those clinically relevant latent factors, I fit a Cox Proportional Hazards regression, and computed Harrell's c-Index (Pencina and D'Agostino, 2004) (See Methods). The c-Index is a measure of prediction accuracy for censored data, with a score of 0.5 being expected by random guessing, and a score of 1.0 being perfect. *maui* (c=0.72) outperforms both MOFA (c=0.68) and iCluster+ (c=0.64) in this benchmark (Figure 3.3E).

The CMS subtyping scheme, as well as much of the work in the field, is based solely on gene expression profiles. In order to examine whether *maui* gives better predictions of patient survival with the addition of mutations and copy number data, I also trained a *maui* model based on gene expression alone. The *maui* model based on expression alone (c=0.69) achieves a lower score than a *maui* model with multi-omics data (c=0.72), even when the former is given more genes as input features (Figure 3.3F), indicating that data other than transcriptomes do contribute to overall *maui* performance. One of the advantages of *maui* over other methods such as iCluster+ and MOFA is that it is able to learn orders of magnitude more latent factors, at a fraction of the computation time (Table 3.2). In order to demonstrate the advantage of being able to fit larger models, I also trained a *maui* model based on 6,000 multi-omics features (see Methods), and that model (c=0.75) outperforms the smaller model (Figure 3.3F), demonstrating the clinical utility of learning from more input genes.

Finally, I investigated the applicability of using prior information from protein interaction networks for colorectal cancer subtyping. I and others previously incorporated gene-gene interactions using a method called network-smoothing (see chapter 2 on page 33).

Network-smoothing is done by allowing binary mutation values to diffuse over a gene network, a process which assigns non-zero "mutation scores" rather than binary mutation values, to genes which either have mutations, or interact with mutated genes. I used a gene network defining interactions between genes from the STRING-db (Szklarczyk et al., 2016) database of protein-protein interactions. I applied the *netSmooth* (Ronen and Akalin, 2018a) algorithm (see Methods) to the mutation data prior to passing it to *maui* and computed Harrell's c-Index, as above. Network smoothing mutations further improves the clinical relevance of latent factors learned when integrating multi-omics data (c=0.79), (Figure 3.3G).

A closer examination of the clusters reveals how closely the *maui* clusters resemble the CMS subtypes, and where they diverge. CMS1 is captured by cluster 2, CMS2 is split between clusters 3 and 5, CMS3 is captured by cluster 0, CMS4 overlaps with cluster 4, and cluster 1 is mixed (Figures 3.4A-C). A similar conclusion can be reached based on a set of molecular indicators introduced in (Guinney et al., 2015): CMS1 and cluster 2 show the hypermutated (Figure B.6A), CIMP (Figure B.6B), and microsatellite unstable phenotypes (Figure B.6C). They also have similar mutation rates among TP53, APC, KRAS and BRAF (Figure B.6D), a set of commonly mutated genes in colorectal cancers.

Figures 3.4C and B.6 beg the question of why CMS2 was split into two clusters (3 and 5). In order to investigate whether it is biologically plausible that the CMS group needs to be split into two, I performed a differential expression analysis, identifying marker genes for each cluster. I then ran these lists through a gene set enrichment analysis (See Methods). The top pathways associated with each *maui* cluster are associated with a distinct set of pathways (Figure 3.4D). Specifically, cluster 3 is dominated by TGF-β signaling and leukocyte migration, while cluster 5 is dysregulated in the ErbB, Hippo, and Wnt signaling pathways, demonstrating that these are indeed distinct groups with different molecular

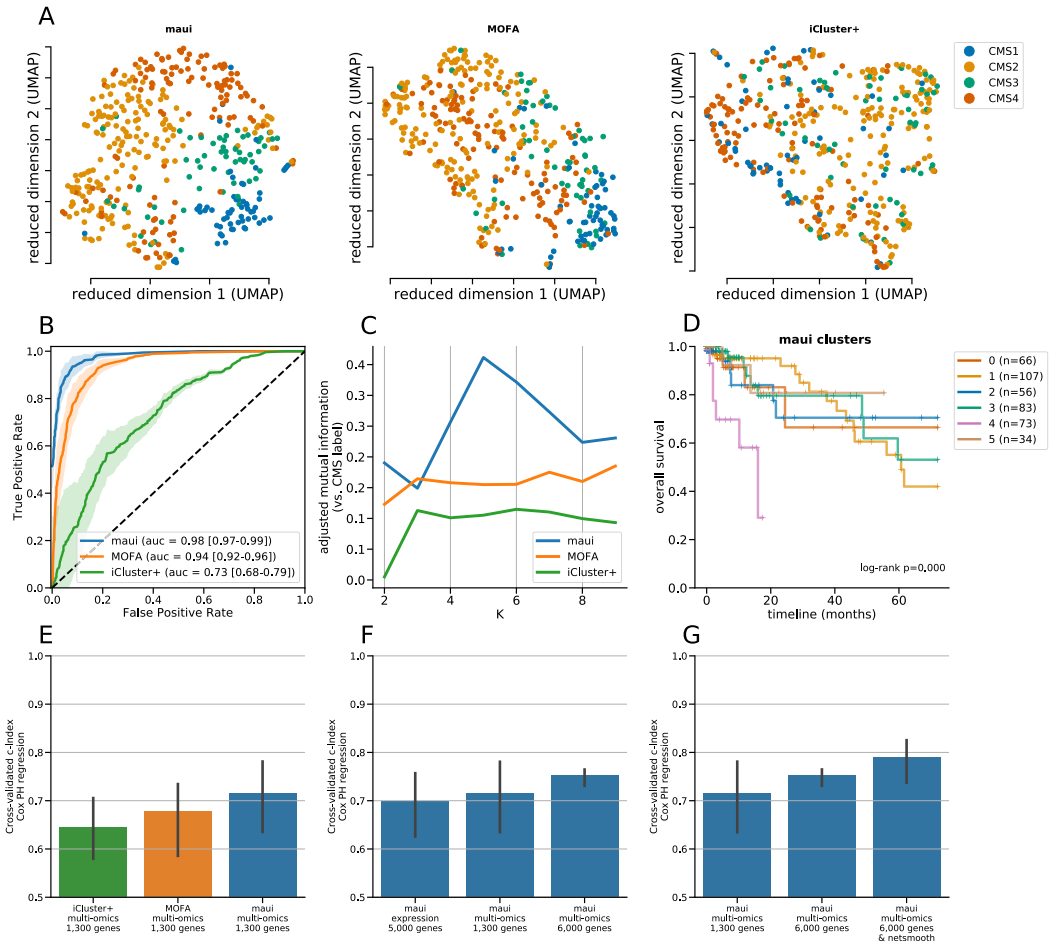


Figure 3.3: maui, MOFA, iCluster+, and the CMS labels. **A)** UMAP (McInnes and Healy, 2018) reduced dimensions from latent factors inferred by maui, MOFA, and iCluster+. Each dot represents a tumor, colored by their CMS label. **B)** ROC's for regularized SVM's predicting the CMS label from latent factors (out-of-sample, 10-fold CV). Mean ROC shown (See Methods) **C)** The Adjusted Mutual Information (AMI, See Methods) of clusters obtained from latent factors inferred by maui, MOFA, and iCluster+, using k-means clustering with K ranging from 2 to 9. **D)** Kaplan-Meier estimates and the log-rank statistic for differential survival of different clusters. The reported P value is from a multivariate log-rank test, under a null hypothesis that all groups have the same survival function. Clusters 3 and 5 represent a novel splitting of a previously defined subtype, CMS2. **E)** Harrell's c-Index for Cox regressions of iCluster+, MOFA, and maui shows maui is more predictive of patient survival than other methods. **F)** Harrell's c-Index comparing different maui flavors shows that maui benefits from multi-omics data, as well as from more input genes. **G)** Harrell's c-Index shows network smoothing of mutations improves survival prediction using maui. This figure is reproduced from Ronen et al. (2018).

phenotypes. Further demonstrating this, cluster 3 presents a worse prognosis than cluster 5 (log-rank $P < 0.001$, Figure B.7). Another cluster, cluster 4 (CMS4) is enriched in pathways associated with mobility and structural differences (Figure 3.4D), which is consistent

with CMS4 displaying more stromal infiltration (Guinney et al., 2015).

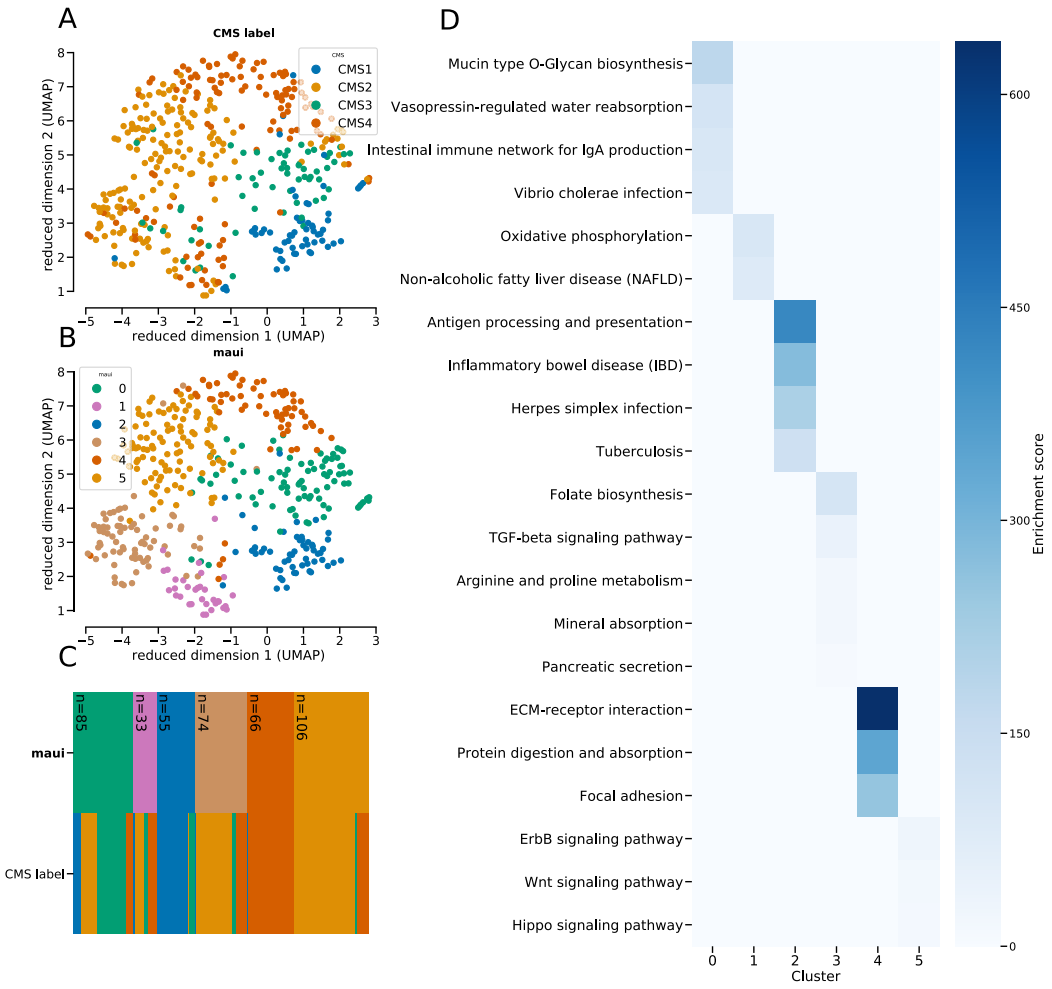


Figure 3.4: Clustering the tumors using k-means using the latent factors from maui reproduces the CMS labels closely, with the exception of CMS2 being split into two clusters, 3 and 5. **A)** UMAP embedding of tumors colored by CMS label, **B)** UMAP embedding colored by k-means clusters on maui latent factors, **C)** Cluster diagram shows the correspondence between maui clusters and the CMS subtypes: the two rows represent the different labeling schemes (maui clusters and CMS subtypes), and each column represents a sample, which is colored according with its assignment in each row. The legend in subfigures A-B applies to the color scheme in C as well. **D)** Pathways that are enriched in differentially expressed genes for each maui cluster. Clusters show a disjoint set of dysregulated pathways, underlining the different molecular phenotypes which underlie each group. Cluster 3 and 5 (which together make up the bulk of CMS2) are dominated by dysregulation of TGF-β signaling, and ErbB/Wnt/Hippo signaling, respectively. This figure is reproduced from Ronen et al. (2018).

Method	Notes	# Factors	Runtime
iCluster+	Bayesian, MCMC	10	∼11hrs
MOFA	Bayesian, variational	20	20mins
maui	*Multilevel Bayesian, Stochastic Gradient Descent*	*100*	*3 mins*

Table 3.2: Summary of methods. This table is reproduced from Ronen et al. (2018).

CRC latent factors are associated with processes related to tumour progression and development

Thanks to its superior computational efficiency, *maui* is able to infer many latent factors from multi-omics data. This creates an opportunity to select the most interesting latent factors and treat them as potential biomarkers. In order to demonstrate this, I fit Cox Proportional Hazards models (Breslow, 1975), fitting one regression model for each factor, as above, selecting clinically relevant latent factors (See Methods). Figure 3.5B shows the 95% confidence interval of coefficients for these latent factors, showing that high values for some of these latent factors are predictive of a poor prognosis ($\beta > 0$), while others are predictive of more favorable outcomes ($\beta < 0$). In general, these latent factors can be used as biomarkers with a significant prognostic value.

Beyond the potential to use these latent factors values as biomarkers in order to prognosticate, it is important that I be able to interpret what these biomarkers represent. *maui* is very powerful because it can learn highly non-linear patterns. This comes at a certain cost: the biological interpretation of the factors is less straightforward than in a linear matrix factorization approach, like PCA or MOFA. PCA and MOFA learn linear relationships be-

tween genes and latent factors, of the form $x = Wz$, where W is directly available, and in it the connections between latent factors and genes. *maui* does not produce a straightforward, linear W, and so, in order to associate latent factors with input genes, I correlated input genes with latent factor values (see Methods). While most latent factors are active in the gene expression domain, most are not significantly affected by mutation data, while others capture interactions between two or more omics types (Figure 3.5A). By correlating latent factors with input features in this way, we can overcome the difficulties presented by the nonlinear relationships between factors and input features, and use the associations in order to find biologically relevant interpretations for neural latent factors.

When I associated clinically relevant (See Methods) latent factors with gene ids, I observed enrichment of pathways known to play a role in CRC such as Wnt signaling and other APC mediated processes (Figure 3.5C). In addition, one of the the most significantly survival associated factors is enriched in Neuronal growth factor (NGF) signaling associated genes. NGF signaling, which controls neurogenesis, is associated with aggressive colorectal tumours (Jobling et al., 2015; Liebig et al., 2009). Survival-relevant latent factors also implicate Platelet-Derived Growth Factor (PDGF) signaling which is also associated with stromal invasion and poor prognosis for colorectal cancer patients (Kitadai et al., 2006; Steller et al., 2013). Thus, in addition to using latent factors as potential biomarkers for prognosis, we can also point at the underlying biological processes that are uncovered by *maui*, potentially driving future drug-target studies.

Quality assessment of CRC cell lines as models for tumors

The molecular profiles of cancer cell lines often differ significantly from those of tumors, due to the differences in selective pressures faced by cells in culture and natural tu-

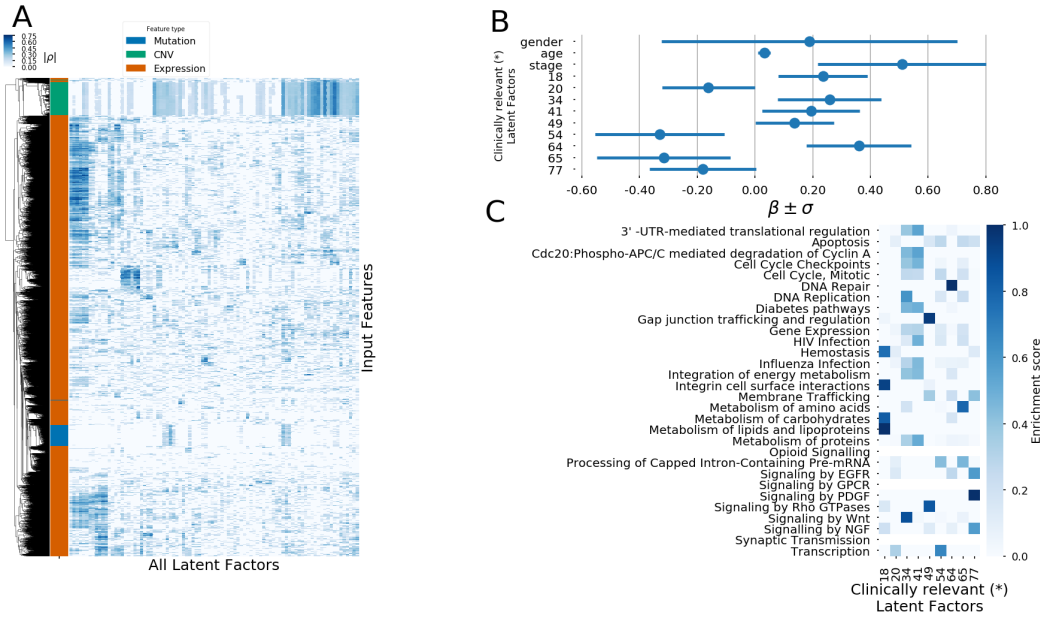


Figure 3.5: Interpretation of maui latent factors. **A)** A heatmap showing the absolute correlation coefficients of the different input genes with the latent factors. Only input features with significant correlations ($P_{adj} < 0.01$, see Methods) are shown in the heatmap. The row annotation shows the type of input feature, i.e. expression value, mutation, or copy number. **B)** The coefficients in a Cox Proportional Hazards regression for factors which are clinically relevant (*) when controlling for patient age, sex, and tumor stage. Coefficients also shown for those covariates. **C)** Pathway enrichment scores for genes associated with the latent factors which carry prognostic value (have significant effects in Cox regression). **(*)** Clinically relevant factors are factors with a coefficient in a fitted Cox model controlling for age, sex, and tumor stage, which are statistically significantly nonzero ($P_{adj} < 0.05$). This figure is reproduced from Ronen et al. (2018).

mor microenvironments; adaptation requires distinct genomic alterations Ben-David et al. (2018). This means that not all colorectal-derived cancer cell lines are likely to have equal value as models for tumors. Furthermore, over time cancer cell lines run the risk of contamination and mis-labeling. For instance, a cell line which was originally annotated as colorectal, has been shown to be derived from melanoma Medico et al. (2015). Since the identification and quality control of the cell lines are crucial steps in the research process, it is essential to know if the lines have diverged too much from tumors in their molecular makeup, been mis-labeled, or contaminated. We examined 54 cancer cell lines derived from tumors of the colon from the Cancer Cell Line Encyclopedia (CCLE). We used maui to

infer latent factor values for the cell lines, to permit their characterization using the same latent factors as the tumors. As cell lines may develop adaptations specific to cell culture, their molecular profiles are often more similar to other cell lines than to those of primary tumors. We therefore hypothesized that cancer cell lines that are more similar to other cell lines than to tumors are less likely to be appropriate models for CRC tumors. We compiled a list of nearest neighbors (See Methods) for each cell line, and then counted how many of its nearest neighbors are cell lines (as opposed to tumors). We used euclidean distance in the space defined by the latent factors to determine similarity, and found that about half of the colorectal cancer cell lines we investigated belong to a "cell line cluster", meaning that the majority of their neighbors were other cell lines (Figure 3.6A). We eliminated cell lines where this proportion is above half, and found among them a mis-labeled cell line: *COLO741*, which has been shown to derive from melanoma and not colorectal cancer[*]. This finding indicates the merit of using this method to flag cell lines as poor models for tumors.

In lieu of knowledge of other mis-labeled or otherwise inappropriate colon-derived cancer cell lines, we artificially contaminated the data set by adding a random sample of 60 non-colon cell lines, assuming that these would be ill-suited to the study of colorectal cancer tumors[†]. We used this to repeat the exercise of counting the nearest-neighboring cell lines. With the introduction of these true positives[‡], we found that more of the cell lines could be assigned to a "cell line cluster" in which the majority of their neighbors are other cell lines (Figure 3.6B). For nearly all non-colon derived cell lines, the 5 nearest neighbors were other cell lines, while this was not the case for colon-derived cell lines (Figure 3.6C). As a result, we designated cell lines whose 5 nearest neighbors are other cell lines, as less suit-

[*]In more recent versions of the CCLE annotations, this has been fixed.

[†]The identities of these "known contaminant" cell lines are irrelevant, as we show later that the method works on 100 such random draws.

[‡]Non-colon cancer cell lines are considered true positives in the task of predicting which cell lines are poor models for CRC tumors.

able for the study of colorectal tumors ("rejected"), as they more closely resemble other cell lines, even those derived from other tissues. We retain cell lines with at least one tumor among their 5 nearest neighbors as more likely to be suitable models. The choice of $K = 5$ for the number of nearest neighbors is immaterial, as the method is insensitive to the choice of K (Figure B.8). UMAP embedding of the latent factor space of tumors (with CMS labels, n=419), colorectal cancer cell lines (n=54), and non-colorectal (artificial contamination, n=60) cancer cell lines shows that this procedure eliminates most contamination cell lines, as well as some of the colon cancer cell lines, and that non-rejected cell lines are spread among all clusters (Figure 3.6D). We repeated the analysis with 100 more random draws of 60 additional contaminants. For each such draw, we rejected any cell line whose 5 nearest neighbors are cell lines. This method consistently rejects almost all known contaminants, as well as about half of the colorectal cancer cell lines (Figure 3.6E). Rejecting these cell lines is not necessarily a mistake because even if they originate in colon cancer, this does not guarantee they will be good genomic models for a such tumors, due to e.g. genomic divergence, mis-labeling, or contamination. Additionally, the fact that a particular cell line more closely resembles non-colon-derived cancer cell lines than CRC tumors is an indication that it might not be suitable as a model for colorectal cancers. That this method successfully rejects almost all known contaminants is another indication that rejected colon cancer cell lines are likely to be poor models for CRC as well. The colorectal cancer cell lines CL40, SW1417, and CW2 are deemed most suitable as models for CRC tumors (Figure 3.7). Using the same criteria, the cell line COLO320 ranked among the lowest. COLO320 lacks mutations in major CRC driver genes such as BRAF, KRAS, PIK3CA and PTEN, and it is actually of a neuroendocrine origin Ahmed et al. (2013); Berg et al. (2017). This very likely makes COLO320 a poor model for CRC.

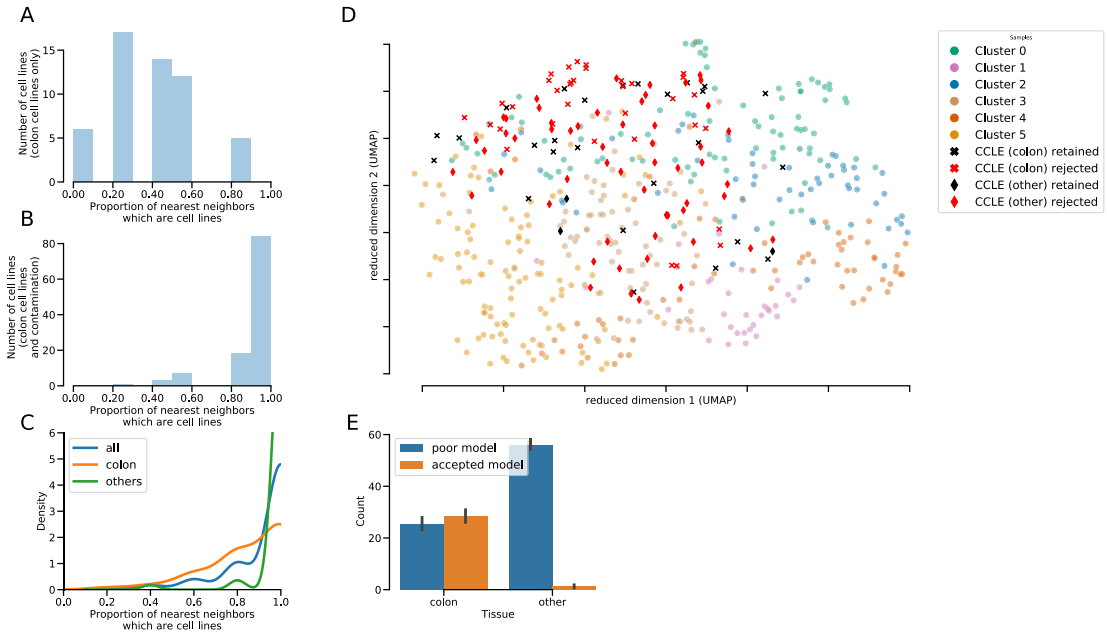


Figure 3.6: For each cell line, we compiled a list of 5 nearest neighbors in latent factor space, and counted the proportion of those nearest neighbors who are cell lines (as opposed to tumors). Cell lines whose 5 nearest neighbors are all other cell lines, are marked as less likely to be appropriate models for tumors, as they are more similar to cell lines than to tumors. **A)** histogram of the proportion of nearest neighbors of cell lines which are also cell lines, colorectal cancers only, **B)**, histogram of the proportion of nearest neighbors of cell lines which are also cell lines, colorectal cancers and non-colorectal cell lines **C)** KDEs of the proportion of cell-line neighbors among all cell lines (colorectal and non-colorectal), broken down by tissue, **D)** UMAP embedding of tumors and cell lines. Crosses are colon-derived cell lines, diamonds are artificial contamination (non-colon derived cancer cell lines). Red cell lines are rejected, black ones are retained as more likely to be good models. **E)** The proportions of colon and non-colon cell lines which are rejected because their proportion of nearest-neighbor-cell-lines is above the threshold. Nearly all non-colon cell lines are consistently rejected, as well as about half of the colon cell lines.

A complete subtyping scheme for CRC and appropriate cell lines for the study of each subtype

The Consensus Molecular Subtyping (CMS) scheme (Guinney et al., 2015) is incomplete as it leaves many tumors without a CMS label. I used *maui* to assign subtypes to the remaining non-CMS tumors by repeating the clustering analysis, and including also tumors that don't have a CMS designation, as well as cancer cell lines. By also including the

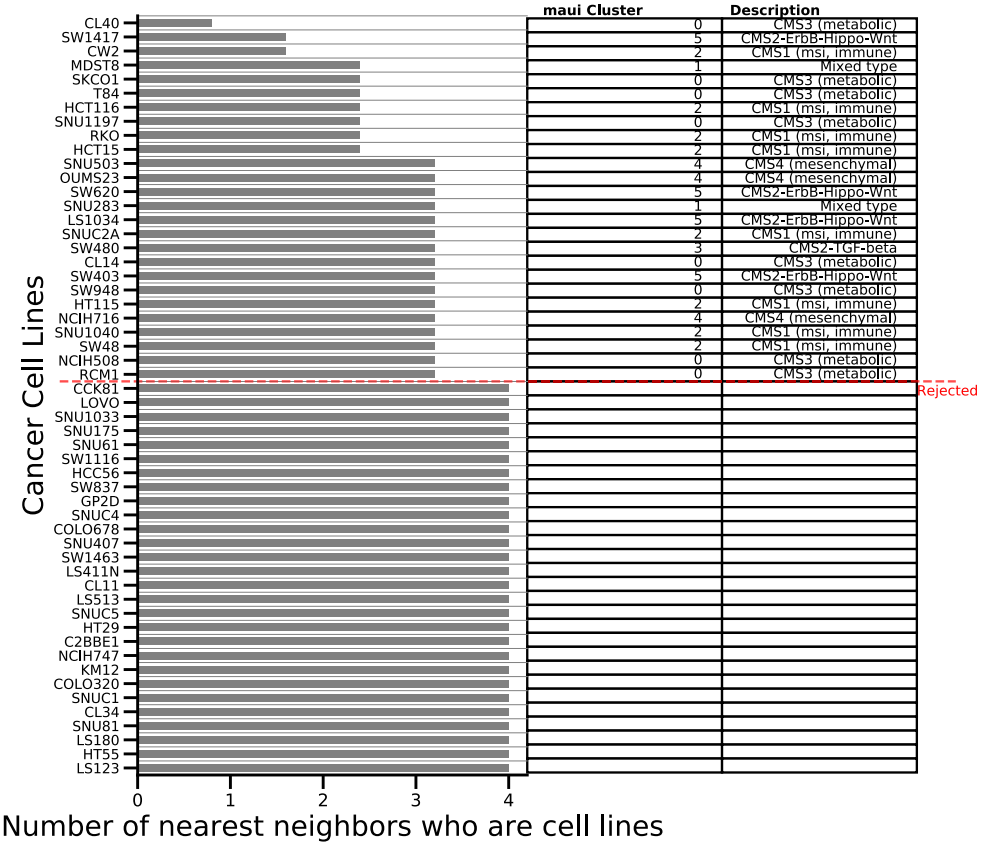


Figure 3.7: For all colon-derived cell lines, we compiled lists of their 5 nearest neighbors. The barplot shows how many of those 5 were other cell lines. Cell lines where all 5 nearest neighbors are other cell lines are rejected, those having at least one nearest neighbor who is a tumor are kept and assigned to clusters, as shown in the table on the right.

cancer cell lines which were deemed to be suitable models (see above), I also assigned CRC subtypes to the cancer cell lines. Here, I present a novel subtyping scheme for CRC, which covers the whole TCGA cohort (including tumors without a CMS designation), as well as an association of CRC cell lines with these subtypes. The non-CMS samples are distributed roughly according to cluster size, as is to be expected for samples that lack a consensus definition (Figure 3.8A-B), and all clusters have at least one associated cell line (Figure 3.8C, Table 3.3). The cluster with the most associated cell lines is cluster 2 (CMS1, MSI), which consists of hyper-mutated tumors with low chromosomal instability, and indeed the cell

lines that I matched with cluster 2 show the same characteristics (Figure B.9), again indicating that latent factors capture patterns which are important to cancer biology. I hope that this can be a useful resource for future drug discovery studies in colorectal cancers.

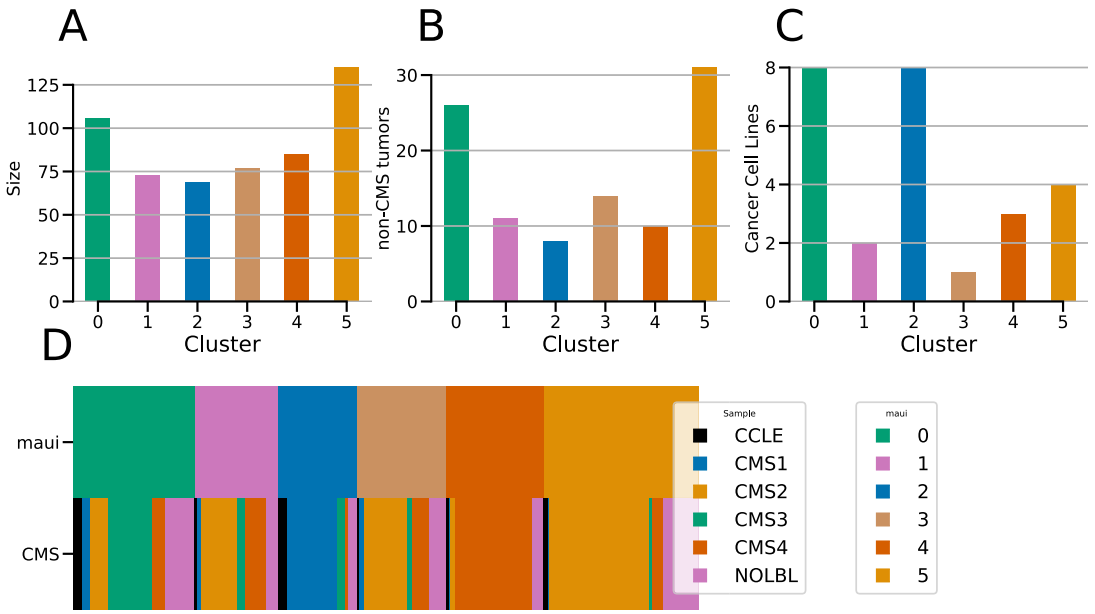


Figure 3.8: A) The sizes (number of samples) of the clusters, **B)** The number of non-CMS tumors assigned to each cluster, **C)** the number of cancer cell lines associated with each cluster **D)** Cluster diagram shows the correspondence between maui clusters and the CMS subtypes: the two rows represent the different labeling schemes (maui clusters and CMS subtypes), and each column represents a sample, which is colored according with its assignment in each row. The NOLBL samples without a defined CMS subtype are distributed among all clusters, as are cancer cell lines (CCLE). This figure is reproduced from Ronen et al. (2018).

3.4 Discussion

Colorectal cancer (CRC) is a heterogeneous disease, with different subtypes being driven by different kinds of genomic alterations, e.g. hypermutated tumors, tumors showing chromosomal instability, etc. Multi-omics data analysis has the potential to increase the understanding of different subtypes of the disease, and new methods which scale

Cluster	Description	Cell lines
0	CMS3 (metabolic)	SW948, CL14, SNU1197, RCM1, NCIH508, CL40, T84, SKCO1
1	Mixed type	SNU283, MDST8
2	CMS1 (msi, immune)	CW2, HT115, SNU1040, HCT15, SW48, HCT116, RKO, SNUC2A
3	CMS2-TGF-beta	SW480
4	CMS4 (mesenchymal)	OUMS23, SNU503, NCIH716
5	CMS2-ErbB-Hippo-Wnt	SW403, LS1034, SW620, SW1417

Table 3.3: maui clusters and the cancer cell lines associated with them. This table is reproduced from Ronen et al. (2018).

computationally are necessary as the amount of available data increases. Apart from stratifying patients into clinically relevant subgroups, it is necessary to find potential drug targets specific to each subtype. Most drug target discovery studies use cancer models such as cell lines, organoids, or xenografts, and it is thus necessary to match these cancer models to the appropriate subtype in each study, or if a cancer model is inappropriate for the study of any subtype, to be able to flag it as such.

I have developed an autoencoder-based method, called **maui**, for integrating data from multi-omics experiments, and demonstrated it using RNA-seq, SNPs and CNVs. *maui* infers latent factors which explain the variation across the different data modalities. The latent factors inferred by *maui* capture important biology such as different gene expression programs, mutational profiles, copy number profiles, and their interactions. I showed that, using *maui* to learn latent factors in multi-omics data, we get latent factors which are predictive of previously described CRC subtypes (the Consensus Molecular Subtypes, CMS). *maui* outperforms the other methods I benchmarked it against, namely iCluster+ and MOFA. *maui* also outperforms MOFA and iCluster+ in survival prediction regardless of the CMS subtypes. From the standpoint of computational performance, *maui* can

extract more latent factors from larger datasets, at a fraction of the computational cost of both iCluster+ and MOFA, making *maui* better suited to the analysis of the larger datasets we expect to see more of in the future. I have shown that *maui* is able to leverage its computational efficiency to learn from larger data sets, containing more genes, to produce latent factors which are more predictive of patient survival. These latent factors also produced a novel classification for CRC. While this novel classification reproduced the CMS nearly perfectly, it revealed that one of the CMS subtypes, CMS2, is in fact two distinct tumor subtypes, with different survival characteristics, and different underlying gene expression programs. These results show that an unbiased selection of more input genes, rather than restriction to known markers or driver mutations, increases prognostic value. Our results support the idea that passenger mutations as well as driver mutations could have an effect on cancer prognosis (McFarland et al., 2017).

iCluster+, which we compared to maui in the first part of this study, is already strained at 1,300 input genes (runtime of 11 hours), and in the future, with even more data types (e.g. methylation), we expect the input spaces to grow far beyond the 6,000 that were used here*. Hence, the computational efficiency of maui is not a mere academic exercise; at today's scale, this increase in computational efficiency is the difference between a model that can be fit at all and one which cannot. In addition to using more input features, the computational efficiency allows maui to learn more latent factors than we might believe to truly exist in the data. This is desirable in latent factor models, as fitting more latent factors increases the amount of ground-truth factors which are recovered by a method. This effect tends to outweigh any harm that may come from overparameterizing the model Buhai et al. (2019). By e.g. ranking latent factors by their clinical relevance (as in Figure 3.5B), we have

*We still recommend that users of maui who are interested in using e.g. DNA methylation data perform some feature selection on the 450k or so CpG's, in a similar way to the feature selection we performed on gene expression, mutation, and copy number data.

shown that we can fish out the ground truth latent factors from a potentially overparameterized model. So here too, the computational efficiency premium offered by maui over iCluster+ and MOFA comes with real-world benefits.

The latent factors can also be individually associated with genes, as well as by their individual relevance to survival prediction. When I performed a pathway analysis on the latent factors which are most predictive of patient survival, I observed enrichment of pathways which are known to play a role in CRC, such as WNT signaling and other APC-mediated processes, NGF signaling, and PDGF signaling (Kuipers et al., 2015). While the association of latent factors to individual genes is not as straightforward using *maui* as it is using matrix factorization methods, the relevance of the implicated pathways is promising. I also proposed a way to use the latent factors learned by *maui* to predict the fitness of cancer cell lines as models for CRC generally, as well as for specific subtypes. In order to address the first question, I hypothesized that cell lines which are poor models for the study of CRC tumors will show higher similarity to other cell lines than to CRC tumors. By including non-colorectal cancer cell lines in the sample and checking if a cell line is more similar to other cell lines than to CRC tumors, we correctly predict that 98% of non-colorectal cancer cell lines are poor models for CRC. The method also predicts that approximately 45% of the colorectal cell lines are poor models for CRC, a prediction which still needs to be validated by new experiments, although the method reliably rejected previously known inappropriate cell lines such as COLO741 and COLO320 (Medico et al., 2015; Ahmed et al., 2013; Berg et al., 2017). The rejected cell lines may still be used to study genetic interactions etc., but their utility in studies of e.g. adaptive drug response may be limited. On the other hand, SW480 and SW620 cell lines that are predicted to be a good match for CRC show similar drug response to clinical trials on KRAS mutant tumours (Sun et al., 2014).

By including the predicted appropriate cell lines in the clustering analysis, I assigned

CRC subtype-specific cell lines, a finding with far reaching potential for subtype specific drug trials. One of the clusters (cluster 2, CMS1) consists mainly of hyper-mutated tumors with low CIN, and the cell lines I matched with that cluster using *maui*, share those same characteristics; matching such characteristics has been a standard way to find disease-specific cell lines (Cheng et al., 2018), and this shows that *maui* cell line matches also preserve this desired behavior. I hope in the future it can be tested whether our approach to predicting fitness of cancer cell lines as models for tumors can be verified, and extended to other cancer models, such as organoids and xenografts. In that way, *maui* could become an indispensable part of drug discovery pipelines and speed up new therapeutics.

The CRC subtypes I used as a starting point for this study were defined based on gene expression profiles alone. As I wanted to use multi-omics data to refine these subtype definitions, I was limited to a subset of the tumors used in the CMS definition. I used only samples from the TCGA which had measurements for both gene expression, mutations, and copy numbers (n=519), while the CMS study used a larger cohort (n=4,151) and only gene expression profiles. Consequently, it is unclear whether the splitting of the CMS2 subtype into two clusters which I have proposed above would hold when presented with a larger dataset. Only once a larger multi-omics dataset is available will this question be answered.

While the autoencoder architecture of *maui* is able to do inference in larger data at a fraction of the time compared with matrix factorization methods such as MOFA and iCluster+, the resulting model is more challenging to interpret biologically, i.e. linking genes with latent factors is not as straightforward as in matrix factorization. I have proposed to solve this by using correlations of the input genes and the latent factors, picking the most significant ones heuristically. While I was able to show that such latent factor—gene relationships capture meaningful cancer biology and recapitulate known associations between

dysregulation of certain pathways and patient survival, this method is potentially less robust than matrix factorization to these associations, and might require more user involvement in the analysis pipeline.

In this study I have developed a deep learning based multi-omics integration method (*maui*) and shown that it can be used to define clinically relevant subtypes of CRC, as well as predict the fitness of cancer cell lines as models for the study of tumors, and an association of cell lines to CRC subtypes. The latent factors inferred by *maui* are also interpretable in biological context, and predictive of patient survival, which enables the associations between underlying oncogenic processes, and patient survival. I benchmarked *maui* against two state-of-the-art methods for multi-omics data integration, and showed that not only is it more effective in defining clinically meaningful subtypes, it also does so with superior computational efficiency. Being orders of magnitude faster will enable *maui* to be used in studies with larger cohorts and more omics types, as these experiments become more abundant in the future. Further, *maui* is a general tool for multi-omics integrations, and may be used outside of the cancer context as well, in basic biology studies employing multiple genomic assays.

4

Discussion

This doctoral work has resulted in a book chapter about multi-omics data integration in the R programming language, as well as two articles describing original methods for integrative data analysis, one for single cell RNA sequencing data imputation, and another using deep learning to integrate multi-omics data in a joint latent factor model. The methods presented in those publications (and this dissertation) make a contribution to the wide field of computational methods integrating data from different experiments, a field which

is likely to continue growing as experimental protocols for same-sample multi-omics proliferate and the cost of sequencing continues to plummet. The tools have been released to the scientific community under free, open source software licenses, and packaged versions have been made available through standard channels (Bioconductor and the Python Package Index, PyPI), where they have received thousands of downloads[*†].

4.1 Network diffusion to integrate data from past experiments

Single cell RNA sequencing has provided transcriptome profiling at a resolution and throughput which were not possible before, but is often associated with the cost of reduced quality measurements. This reduced quality manifests itself in higher technical variance, which leads to the so-called drop-out phenomenon, where genes which are expressed in a cell are perceived as not being expressed, due to e.g. loss of RNA material while handling such minute quantities (Pierson and Yau, 2015). Drop-outs, i.e. false zero's, have been dealt with in the literature by imputation methods which essentially guess the true expression value of a gene by looking at its expression values in other cells which are similar, e.g. (van Dijk et al., 2017; Lin et al., 2017; Li and Li, 2017).

In chapter 2 I described *netSmooth*, a quasi imputation method I developed based on an orthogonal approach: network diffusion on a gene-gene network. Rather than guess a gene's true expression level based on its expression in other similar cells, my method guesses its expression level based on the expression levels of other genes which are known to be co-expressed. I demonstrated this using a protein-protein interaction network, which is reasonably predictive of co-expression (Bhardwaj and Lu, 2005; Fraser et al., 2004). Using

[*] http://bioconductor.org/packages/stats/bioc/netSmooth/
[†] https://pepy.tech/project/maui-tools

netSmooth, down-stream analysis tasks such as clustering, are improved in biological systems including embryonic development, hematopoiesis, and cancer.

This is an interesting result also because the gene network I used in all three cases was the same — a global protein-protein interaction network of the most confident gene interactions in the STRINGdb database (Szklarczyk et al., 2017). Global in this context means the network represents gene all gene interactions, without filtering for a specific context (e.g. gene interactions which are known to be important in hematopoiesis or other biological systems). However, we know that gene interactions can be cell type specific. Interactions between miRNAs and their target mRNAs follow cell type specific patterns, (Sood et al., 2006), as do protein-protein interactions (Gora et al., 2010). It is a testament to the true signal-recovering abilities of *netSmooth* that it works with global (not context specific) gene networks. Future efforts to infer context specific gene regulatory networks, miRNA–mRNA interactions, and protein-protein interactions, will surely result in an even more effective *netSmooth*.

Recently, it has been suggested that the drop-out phenomenon, i.e. that genes that are expected to be expressed are not detected in scRNA-seq, is not a technical issue, but rather an expected result of the biological variability in gene's transcription rates (Svensson, 2019). It is interesting to observe that the direct imupation methods scImpute, MAGIC, etc. reduce this drop-out phenomenon much more than does *netSmooth* (Figure A.7 on page 119). A consequence of this aggressive imputation is that after applying MAGIC or scImpute, many more gene pairs are significantly correlated than what is reasonable to expect (Figure 2.2 on page 50, Figure A.5 on page 117, Figure A.8 on page 120). In fact, in many cases, MAGIC and scImpute processing leads to gene pairs being highly correlated even when they are never observed in the same cell! (Erik van Nimwegen called this "hallucinating correlations" (Unpublished)). Hence, *netSmooth*'s approach of inferring expression values

based on expected co-expression rather than forced imputation under strong assumptions, is more appropriate for recovering the true signal in scRNA-seq experiments.

These properties make *netSmooth*, the R package I developed implementing this algorithm, a versatile tool for denoising any genomic data for which a high quality gene network may be constructed, in a global or context-specific manner, extending its usability beyond single cell RNA sequencing analysis.

4.2 Using deep learning to integrate multi-omics data

Advances in the mechanistic understanding of tumerogenesis have lead to a refinement of the X-cancer label (where X is some tissue) into more informative sub-diseases. A high-profile example is the sub-typing of breast cancers based on the presence of hormone receptors (reviewed in Althuis et al. (2004)). Targeted therapies developed for the different sub-types (e.g. Herceptin), have lead to a remarkable improvement of both mortality rates and quality of life (due to a reduced used of systemic cytotoxic chemotherapy) (Million Women Study Collaborators et al., 2003). Large consortia such as TCGA have profiled thousands of tumors, including thousands of breast cancer tumors, using different omics platforms, and revealed molecular signatures for the different sub-types which span different omics types (Ciriello et al., 2015). This underlines the need for multi-omics analysis strategies for cancer genomics.

Chapter 3 describes *maui* (Ronen et al., 2018) (**M**ulti-omics **AU**toencoder **I**ntegration), a method I developed, and demonstrates the superiority of the multi-omics-first approach in sub-typing colorectal cancers. *maui* uses state-of-the-art advances in gradient descent (Kingma and Ba, 2014), together with a novel neural network architecture, in order to learn

latent factor representations of multi-omics data with superior computational performance over similar methods. *maui* can integrate multi-omics data from gene expression (RNA-seq), mutations and copy number variations (DNA-seq), in order to learn a latent factor representation for colorectal cancer. Using the latent factor representation produced by *maui*, I was also able to refine the state-of-the-art subtyping system, the Consensus Molecular Subtype (CMS, Guinney et al. (2015)), by splitting one subtype, CMS2, into two separate subgroups with distinct molecular signatures and survival probabilities. The CMS2 should be split into two distinct sub-groups of colorectal cancer, one defined by ErbB, WNT, and Hippo dysregulation, and the other by TGF-β activation. Incidentally, the latter (TGF-β) subtype includes some of the most aggresive tumors in the TCGA cohort, with median survival of $<$ 1 year. The other CMS2 subset (defined by ErbB, WNT, and Hippo signalling activation) makes up some of the least aggressive tumors in the cohort, with median survival of $>$ 6 years (Figures 3.4, 3.3D). Using maui, I was able to answer a question which has been raised by many oncologists in recent times, in fact, each time I presented this work in progress to clinicians — does adding multi-omics data to gene expression actually improve the clinical results? And can whole-genome assays be mined for more clinically relevant information than targeted screens? I was able to answer both questions in the affirmative (Figure 3.3F,G on page 89).

Nearly all of the targeted therapies developed for different cancers rely on cancer models, e.g. cancer cell lines, at some stage in the drug discovery pipeline. A major challenge for drug discovery trial designers is choosing appropriate cell lines to model distinct diseases. Typically, a pharmaceutical company will start by identifying a group of patients which is sufficiently large and lack satisfactory treatment options, to initiate a drug discovery trial. Then, in an initial phase, cancer cell lines which are thought to be good models for the subgroup of interest may be screened for the effects of known compounds, or using genetic

engineering, for critical genes, the silencing of which kills the cell line. Consequently, the identification of appropriate cell lines is a lively field for developments for pharmaceutical companies. I used *maui* to map colon cancer cell lines to the latent factor space defined by colorectal cancer tumors, and assigned each cancer cell line to a subtype, a resource which can help drug discovery trials (Table 3.3). Using this method, I was also able to rank the cancer cell lines by their suitability as models for colorectal cancers, using a novel methodology. The method correctly flagged COLO741 and COLO320 as unsuitable models; these cell lines are known to be inappropriate models for colorectal tumors (Medico et al., 2015; Ahmed et al., 2013; Berg et al., 2017). I also produced a complete ranking of all colorectal cancer cell lines from the CCLE (Figure 3.7).

I benchmarked *maui* against other methods for learning latent factors from multi-omics data, iCluster+ and MOFA, and *maui* outperforms them on the clinical relevance of the latent space it uncovers, and is also orders of magnitude more computationally efficient, which allows learning of factors from many more input features (genes), using many more samples, and at a fraction of the computational cost. Thus, *maui* is well suited for multi-omics integration studies, both in the realm of cancer and in other realms, and I believe *maui* or methods like it will become indispensible parts of many research pipelines in the future.

4.3 Combining netSmooth and maui

In chapter 3, I showed that using *netSmooth* in tandem with *maui* further improves cancer subtyping, generating latent representations of the data with superior clinical relevance (Figure 3.3 on page 89G). This demonstrates the power of computational methods incorpo-

rating priors from the countless experiments which have been made public over the recent decades. As sequencing experiments produce data with high variability, due to both technical and biological reasons, incorporating data from other relevant experiments is a great way to improve analytic results by removing undesired noise and improving pattern recognition.

4.4 Final remarks

Multi-modal artificial neural networks (ANNs) like *maui* have recently been used for protein function prediction by integrating different protein interaction networks (Gligorijević et al., 2018), and for small molecule synthesis (Winter et al., 2019). Also outside the world of computational biology, multi-modal deep learning models have flourished. As a vivid and highly accessible example*, by training an autoencoder-like network using images and text data, with each image being associated with a textual description (caption), such models have been made to generate plausible image captions for previously unseen images (Kiros et al., 2014; Vinyals et al., 2016). With an even stronger hold over our collective cultural imagination of "thinking machines", multi-modal ANNs are behind many recent advances towards autonomous vehicles, where data modalities include (visible light) video, radar, LIDAR, etc. (Chen et al., 2015). More practically, similar models have made breakthroughs in machine translation, by treating text in different languages as multi-modal data (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014); this tool is used by hundreds of millions of people worldwide daily (Wu et al., 2016; Shankland, Shankland). Taken together, these advances, using similar technology in different fields, represent nothing short of a new paradigm of knowledge discovery and rep-

*https://github.com/tensorflow/models/tree/master/research/im2txt

resentation. Another exciting trend in this field, which is best embodied by convolutional neural networks (CNNs) used for image processing, is bio-mimicking of neural networks. CNNs are directly biologically inspired, with convolution followed by pooling layers used in CNNs resembling the LGN–V1–V2–V4–IT structure of the ventral pathway in the visual cortex of a cat (Hubel and Wiesel, 1962; Felleman and Van, 1991; LeCun et al., 2015). Perhaps in the future we will find analoguous biologically inspired models for neural multimodal integration, a task the brain of even much simpler organisms surely performs. Perhaps we will be able to uncover more of biology's greatest tricks for knowledge discovery and representation, and point them directly at biology itself.

A

Supplementary material for Chapter 2

Choice of dimensionality reduction for clustering procedure

netSmooth picks PCA or t-SNE algorithmically, using the 2D entropy in an embedding of a dataset (see section 2.2). For the hematopoiesis and glioblastoma datasets, this is t-SNE, while for the embryonic development dataset it is PCA (Table A.1). This method may be used to pick any dimensionality reduction technique other than the ones mentioned here, which might be more suitable for other analyses.

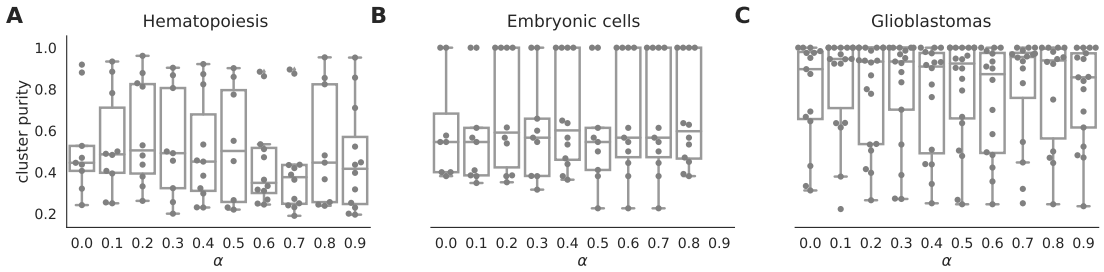


Figure A.1: boxplots of cluster purity for clusters obtained by the robust clustering procedure following application of netSmooth with different values of α. $\alpha = 0$ is equivalent to not using netSmooth at all. The procedure is robust to alpha, that is, most values of alpha produce more robust clusters. A) HSPCs, B) embryonic cells, C) glioblastomas. This figure is reproduced from Ronen and Akalin (2018a).

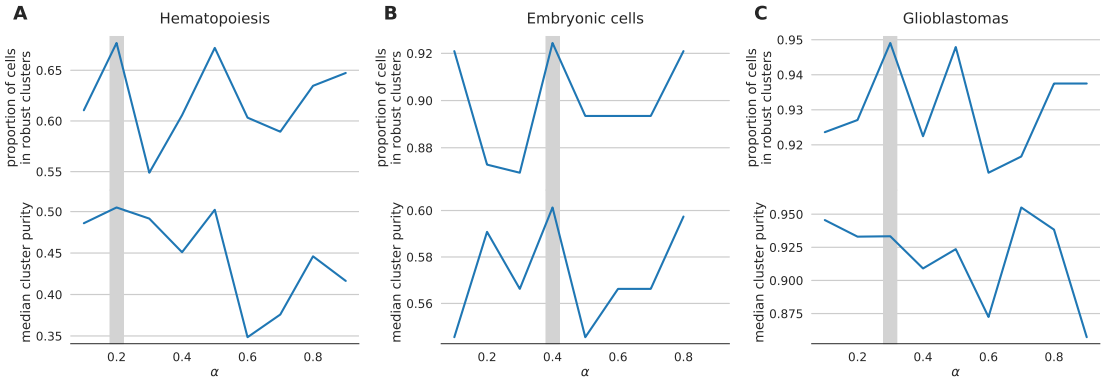


Figure A.2: the proportion of cells in robust clusters, and cluster purity for those robust clusters, for a range of alpha values, shows that picking the alpha with the highest proportion in robust clusters also picks the alpha with the highest cluster purity. A) hematopoietic stem/progenitor cells B) embryonic cells, C) glioblastomas. This figure is reproduced from Ronen and Akalin (2018a).

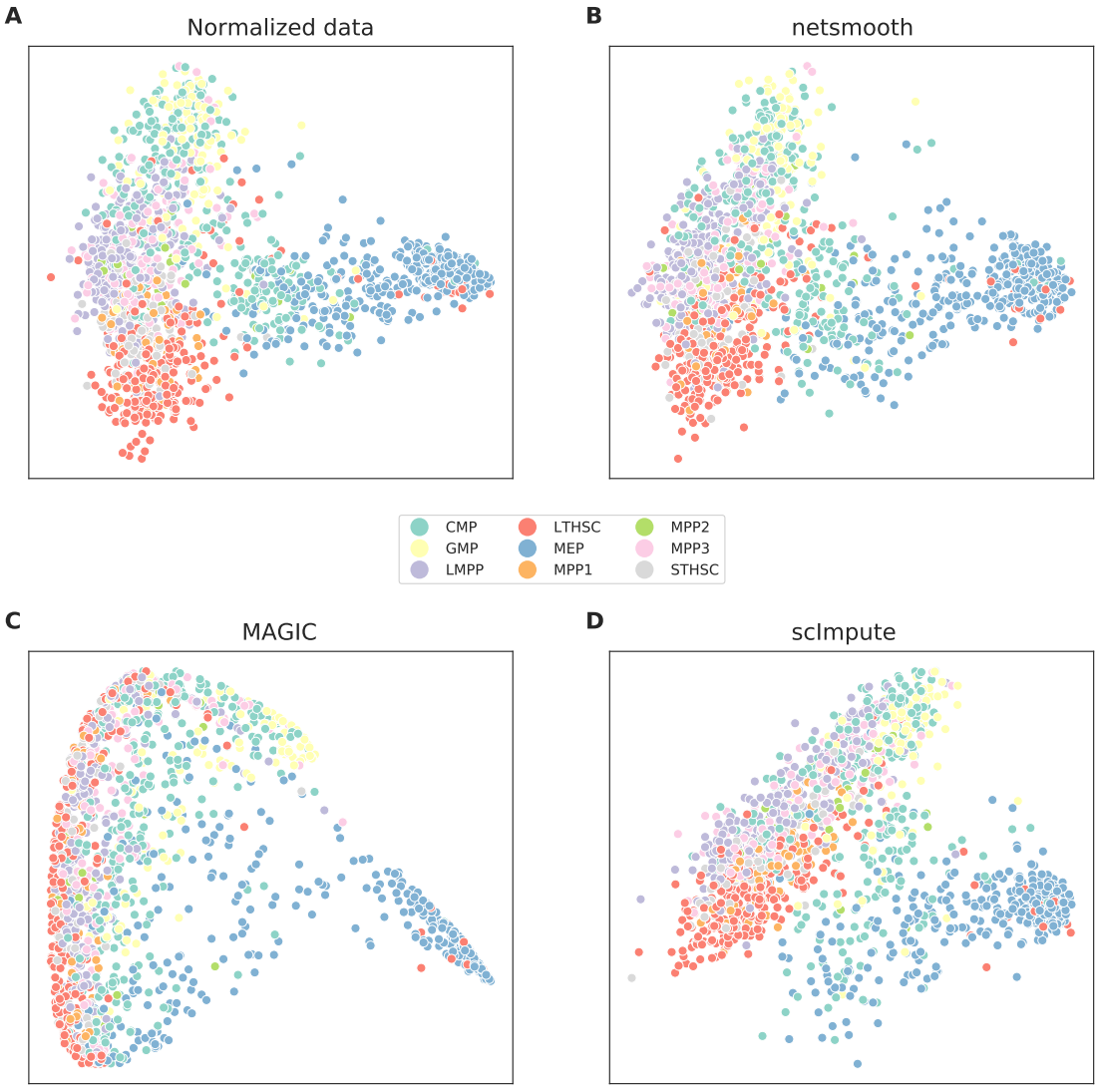


Figure A.3: PCA plots of the HSPC dataset A) no preprocessing, B) after application of netSmooth, C), using scImpute, and D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

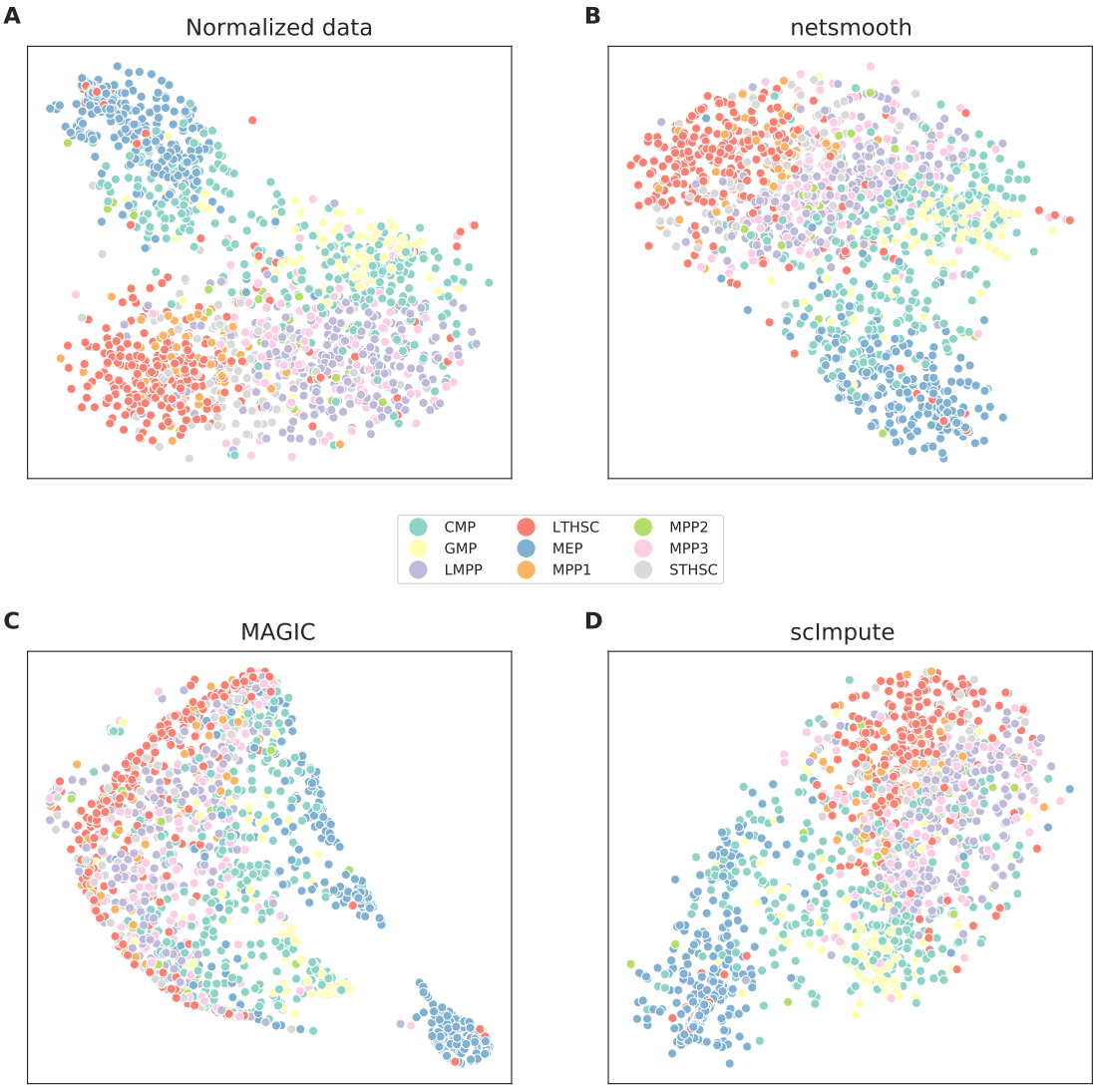


Figure A.4: t-SNE plots of the HSPC dataset A) no preprocessing, B) after application of netSmooth, C), using scImpute, and D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

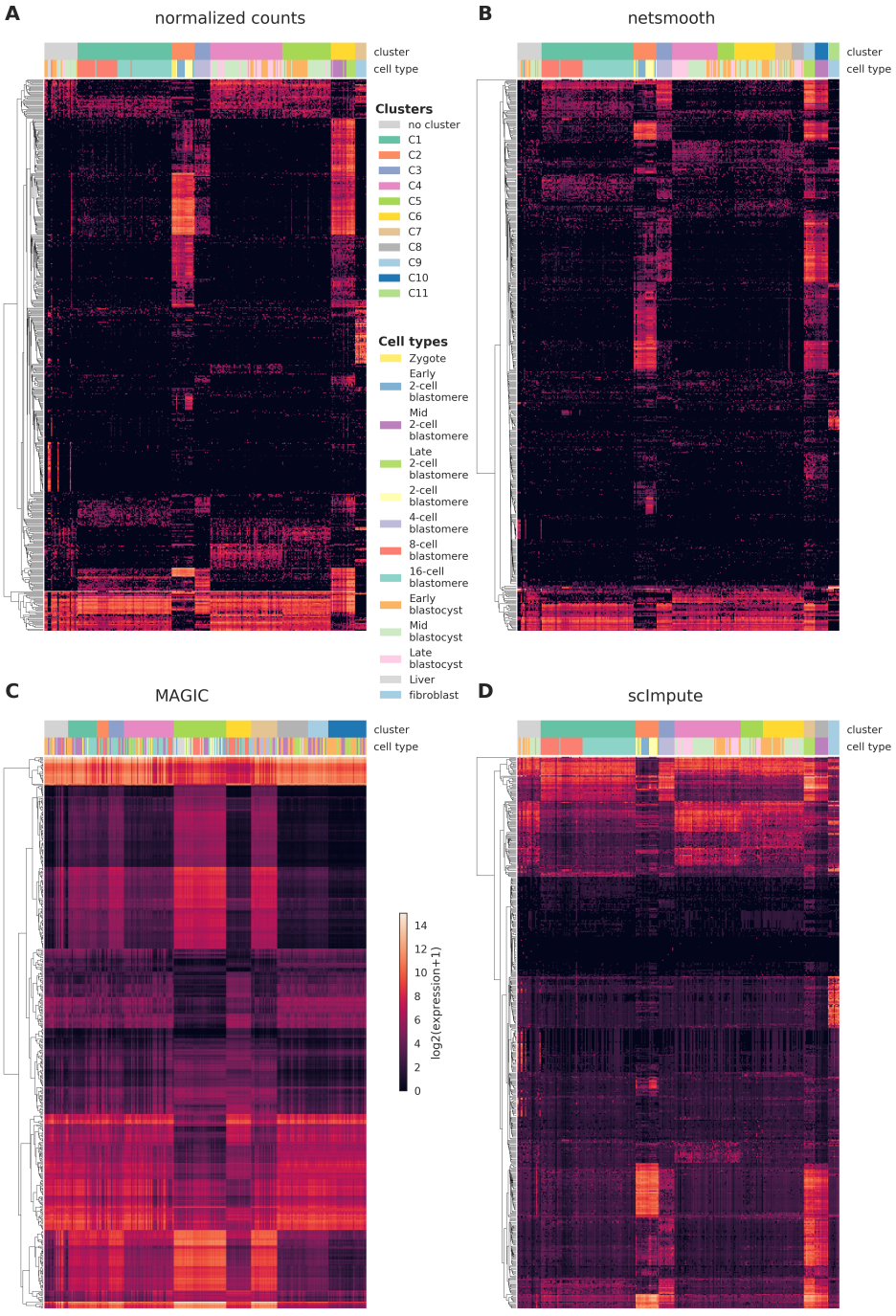


Figure A.5: single cells from the embryonic development dataset were clustered using the robust clustering procedure, and the log-transformed expression values of the 500 most differentially expressed genes (by edgeR-QLF test adjusted P value) in any of the discovered clusters are shown in a heatmap, as well as cluster assignments and cell types. A) raw (no imputation), B) after application of netSmooth, C) missing values imputed using scImpute D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

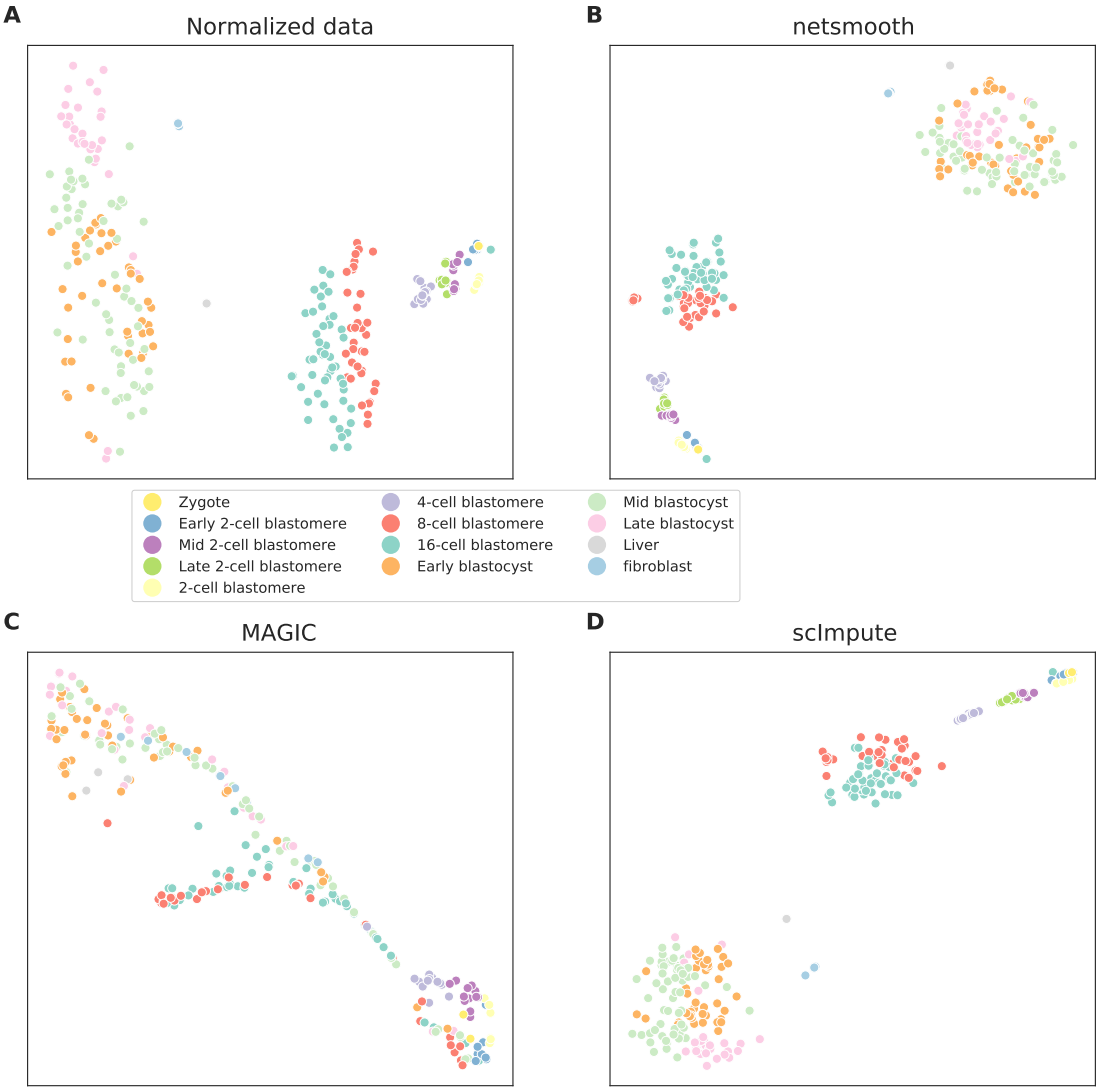


Figure A.6: t-SNE plots of the embvryonic development dataset A) no preprocessing, B) after application of netSmooth, C), using scImpute, and D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

Dataset	PCA Entropy	t-SNE Entropy
Hematopoiesis	4.96	5.03
Embryonic cells	4.09	3.94
Glioblastoma	4.87	5.06

Table A.1: Entropy in 2D lower dimension embeddings. This table is reproduced from Ronen and Akalin (2018a).

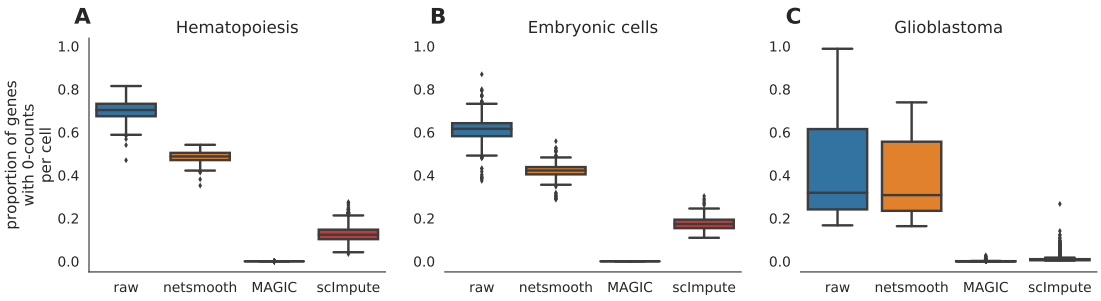


Figure A.7: The proportion of genes with 0 counts is a proxy for technical dropouts. A) no preprocessing, B) after application of netSmooth, C), using scImpute, and D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

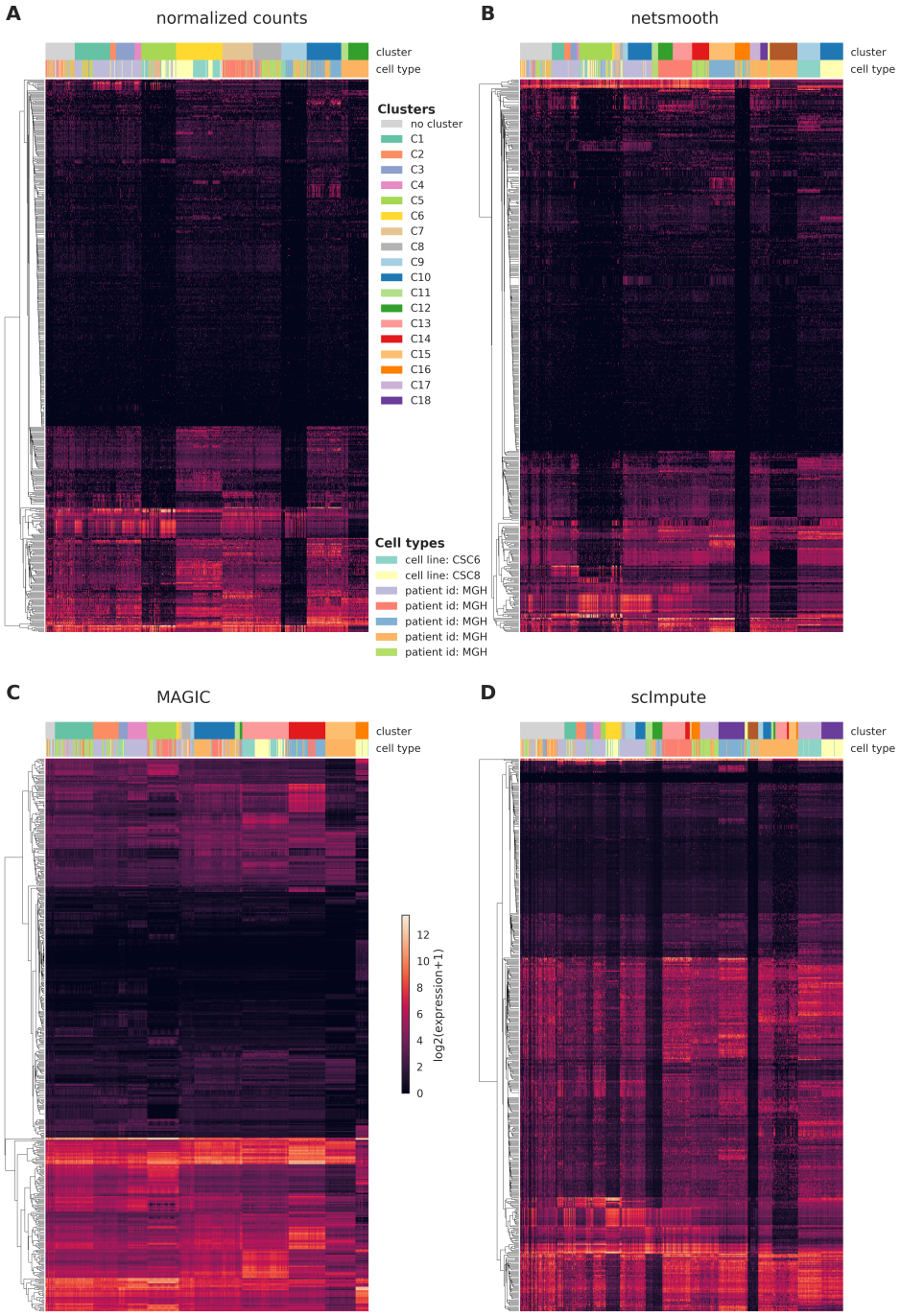


Figure A.8: single cells from the glioblastoma dataset were clustered using the robust clustering procedure, and the log-transformed expression values of the 500 most differentially expressed genes (by edgeR-QLF test adjusted P value) in any of the discovered clusters are shown in a heatmap, as well as cluster assignments and cell types. A) raw (no imputation), B) after application of netSmooth, C) missing values imputed using scImpute D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

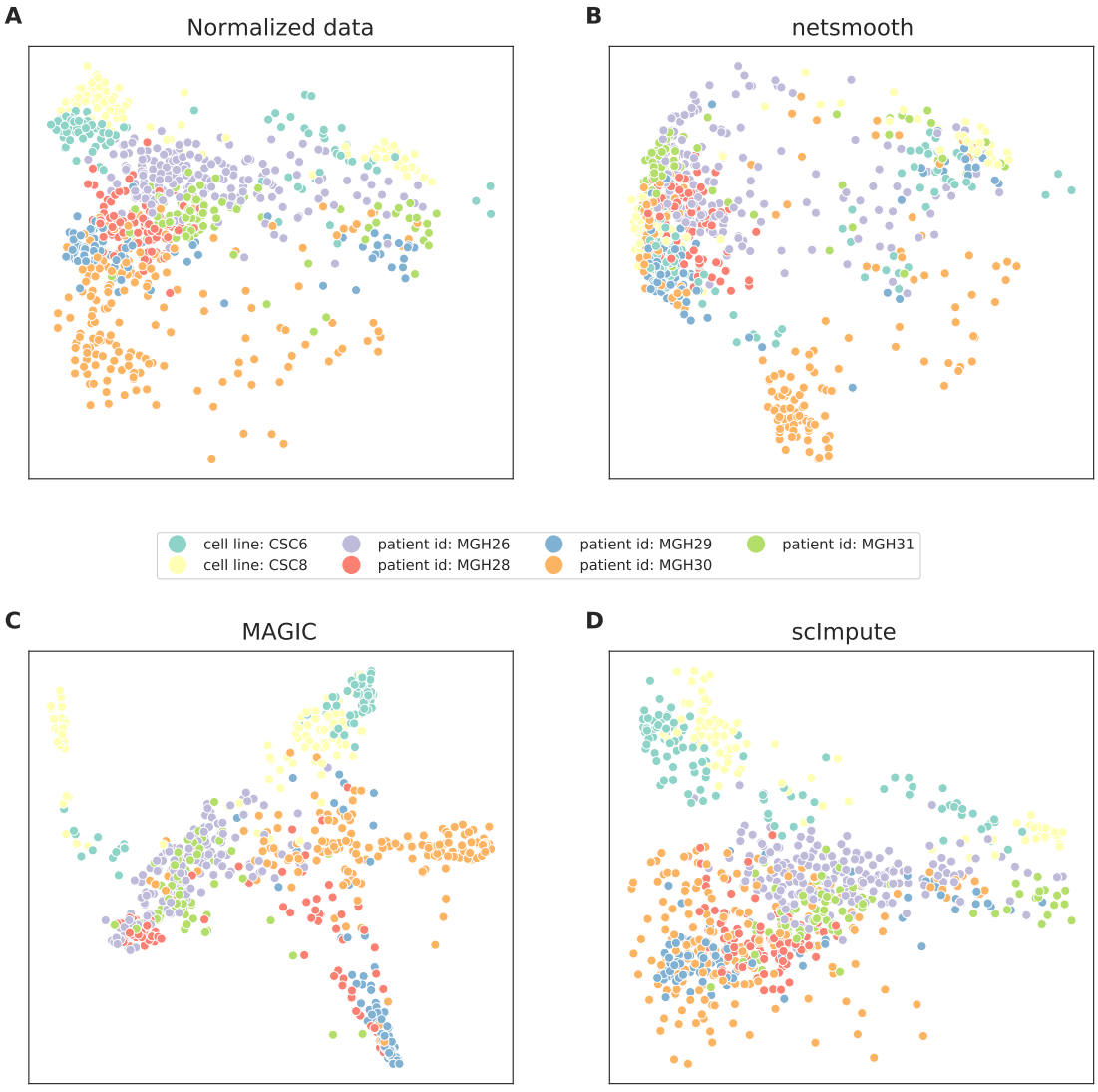


Figure A.9: PCA plots of the glioblastoma dataset A) no preprocessing, B) after application of netSmooth, C), using scImpute, and D) after application of MAGIC. This figure is reproduced from Ronen and Akalin (2018a).

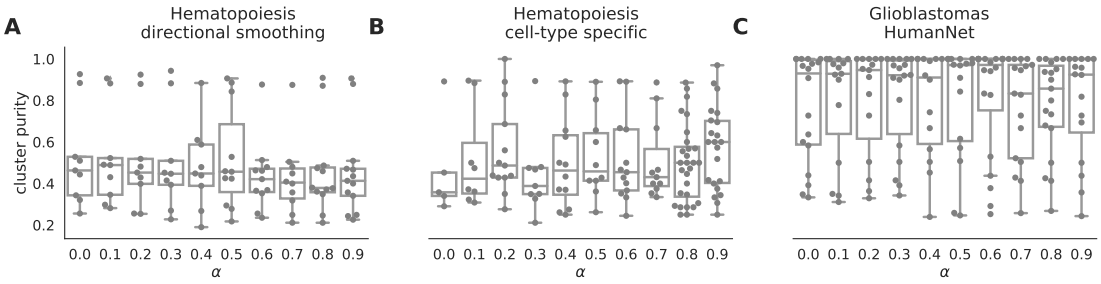


Figure A.10: Cluter purity by smoothing parameter. A) for the hematopoiesis dataset with a directional (signed) graph, where inhibitory interactions have a negative edge weight. B) For the hematopoiesis dataset using a gene network with only genes that have a cell-type specific expression in any cell type. C) In the glioblastoma dataset using a gene network from HumanNet. This figure is reproduced from Ronen and Akalin (2018a).

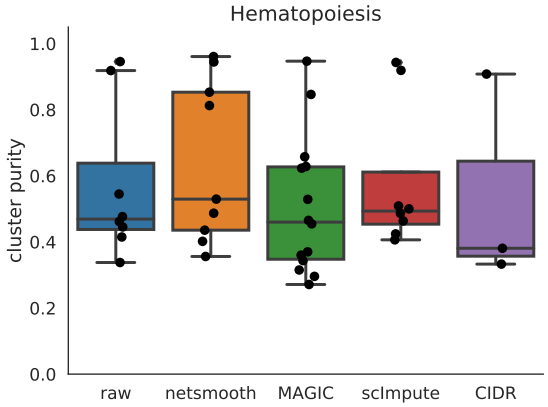


Figure A.11: Cluster purity including CIDR. Same as Figure 2.3, with CIDR included. This figure is reproduced from Ronen and Akalin (2018a).

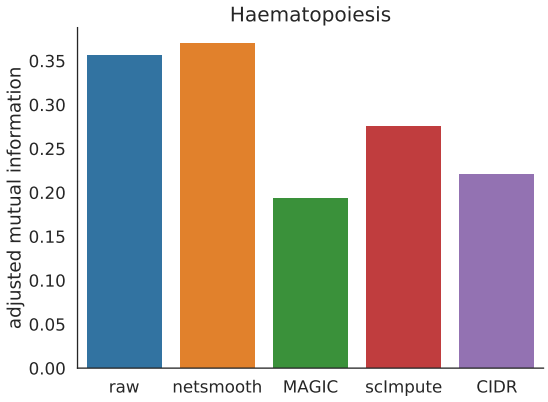


Figure A.12: Adjusted mutual information including CIDR. Same as Figure 2.3, with CIDR included. This figure is reproduced from Ronen and Akalin (2018a).

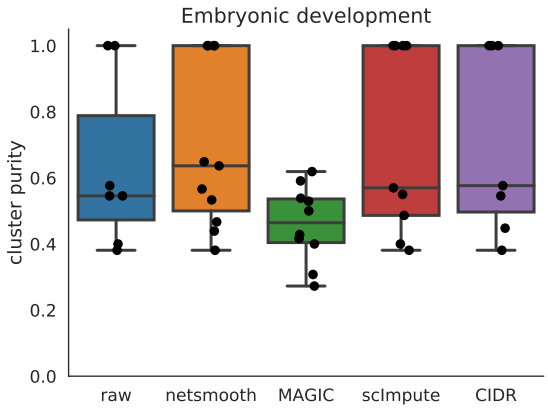


Figure A.13: Cluster purity including CIDR. Same as Figure 2.5, with CIDR included. This figure is reproduced from Ronen and Akalin (2018a).

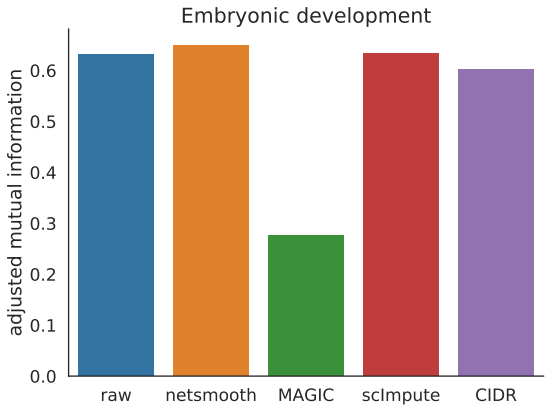


Figure A.14: Adjusted mutual information including CIDR. Same as Figure 2.5, with CIDR included. This figure is reproduced from Ronen and Akalin (2018a).

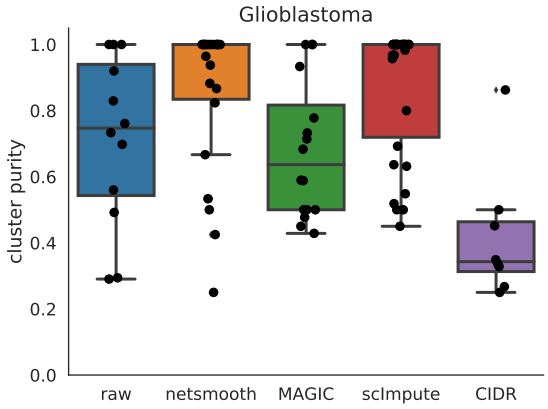


Figure A.15: Cluster purity including CIDR. Same as Figure 2.7, with CIDR included. This figure is reproduced from Ronen and Akalin (2018a).

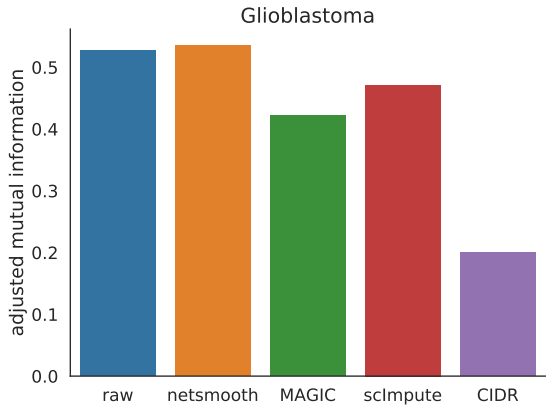


Figure A.16: Adjusted mutual information including CIDR. Same as Figure 2.7, with CIDR included. This figure is reproduced from Ronen and Akalin (2018a).

B

Supplementary material for Chapter 3

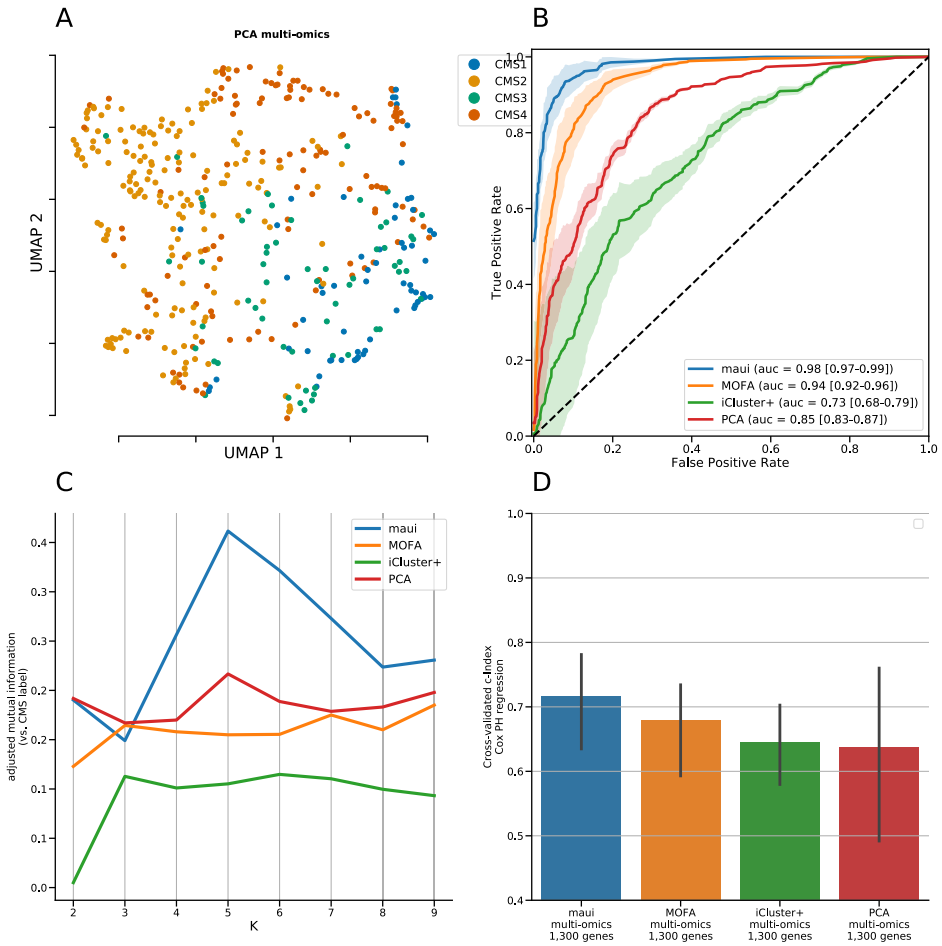


Figure B.1: PCA for multi-omics integrative analysis. **A**) UMAP embedding of PCA of multi-omics fusion. **B**) AUC for prediction accuracy when predicting CMS label from latent factors inferred by maui, MOFA, iCluster+, and PCA. **C**) Adjusted Mutual Information (AMI) between clustering based on latent factors of different methods, and the CMS label, for k-means clustering with a range of Ks. **D**) c-Index of Cox Proportional Hazards model based on latent factors from different methods for multi-omics integration. This figure is reproduced from Ronen et al. (2018).

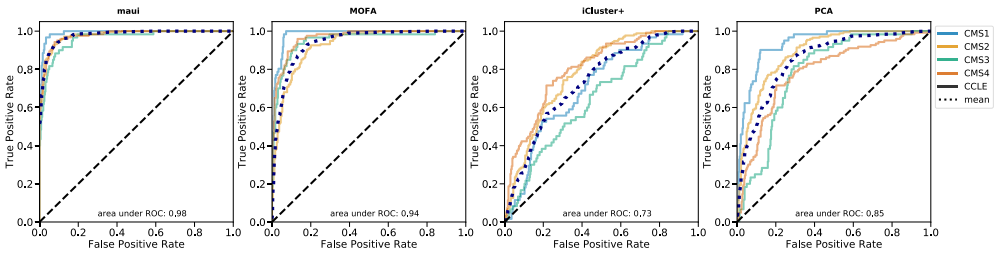


Figure B.2: Receiver Operator Characteristic curves per class (CMS) for maui, MOFA, iCluster+, and PCA. Mean ROC curve also shown. auROC reported is the area under the mean ROC. This figure is reproduced from Ronen et al. (2018).

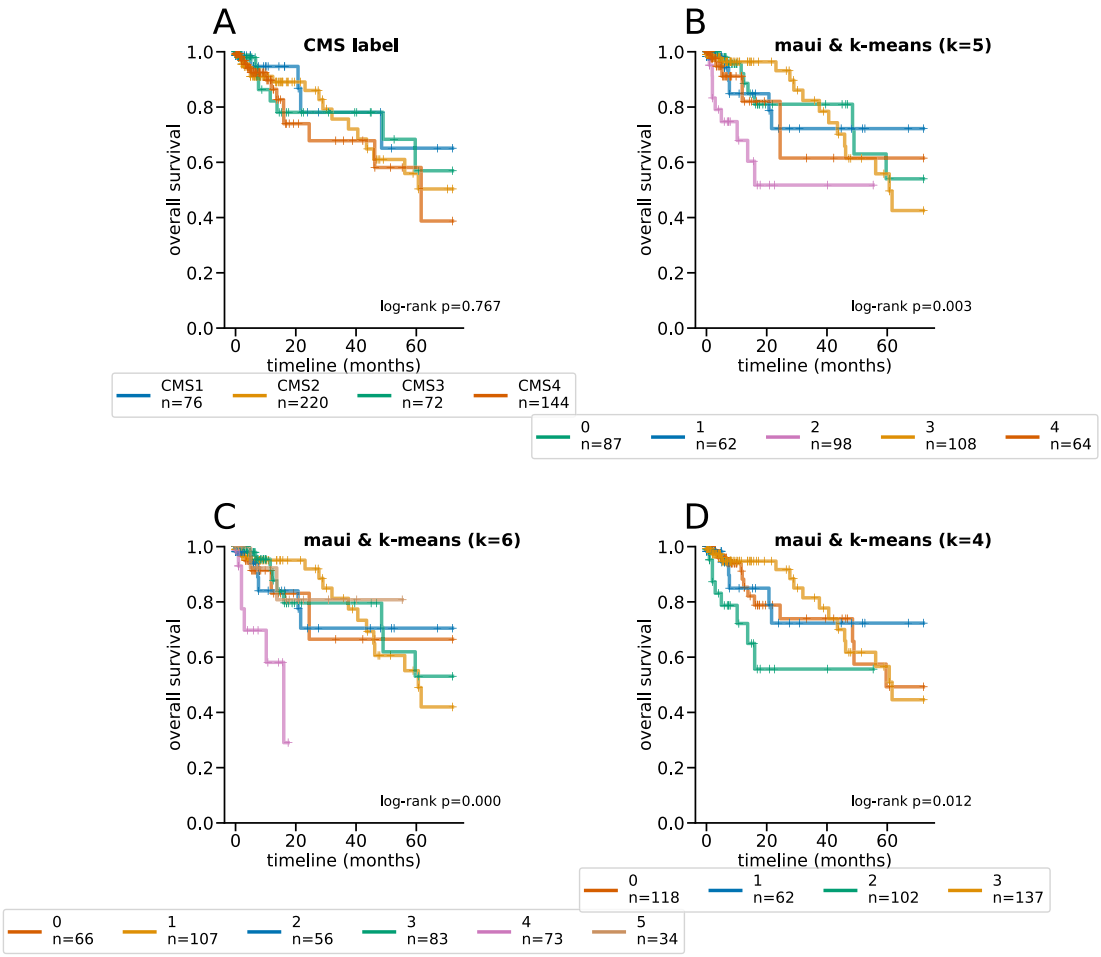


Figure B.3: Kaplan Meier curves and log-rank tests for differential survival statistics for the CMS subtypes, as well as maui clusters using k-means with different K's. The reported P values are from a multivariate log-rank test, under the null hypothesis that all groups have the same survival function. This figure is reproduced from Ronen et al. (2018).

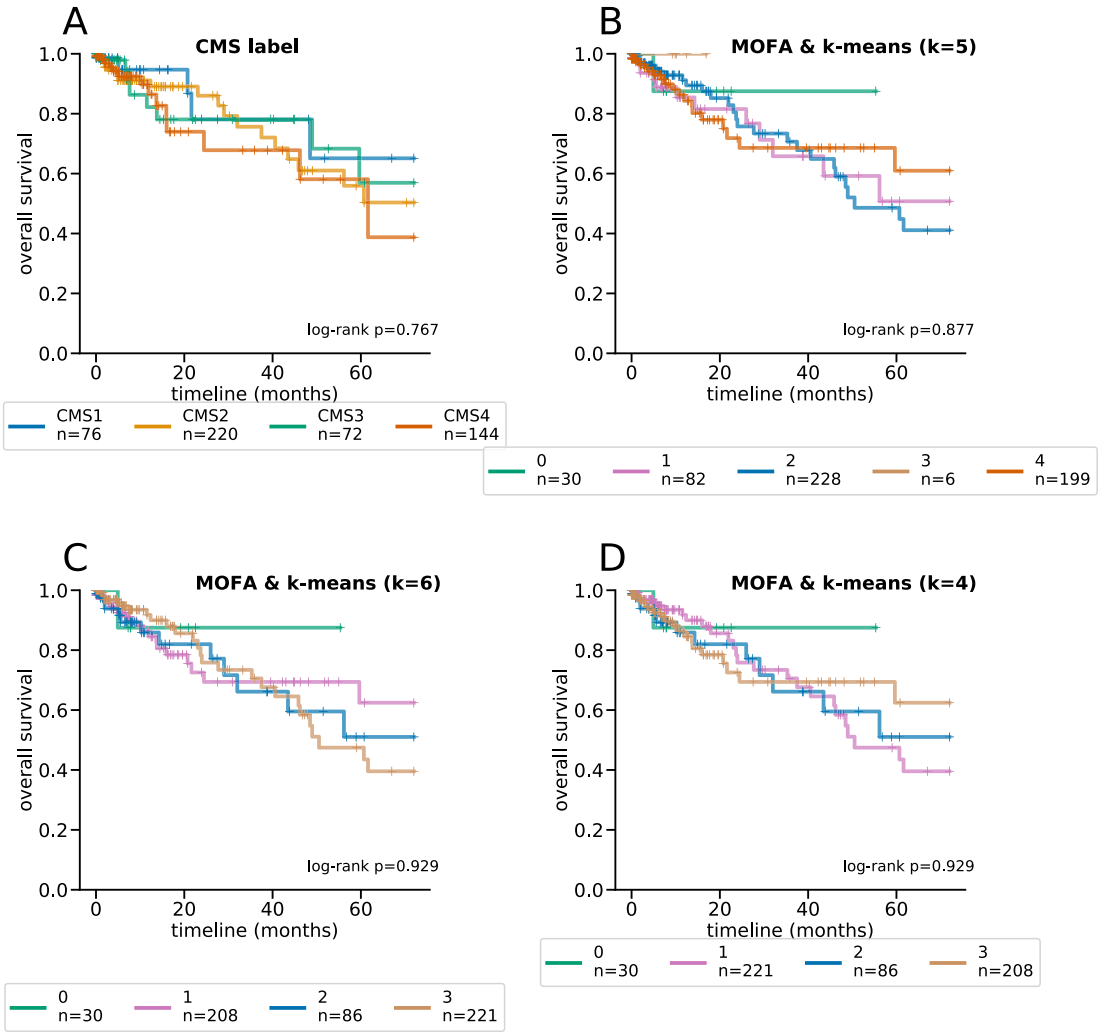


Figure B.4: Kaplan Meier curves and log-rank tests for differential survival statistics for the CMS subtypes, as well as MOFA clusters using k-means with different K's. The reported P values are from a multivariate log-rank test, under the null hypothesis that all groups have the same survival function. This figure is reproduced from Ronen et al. (2018).

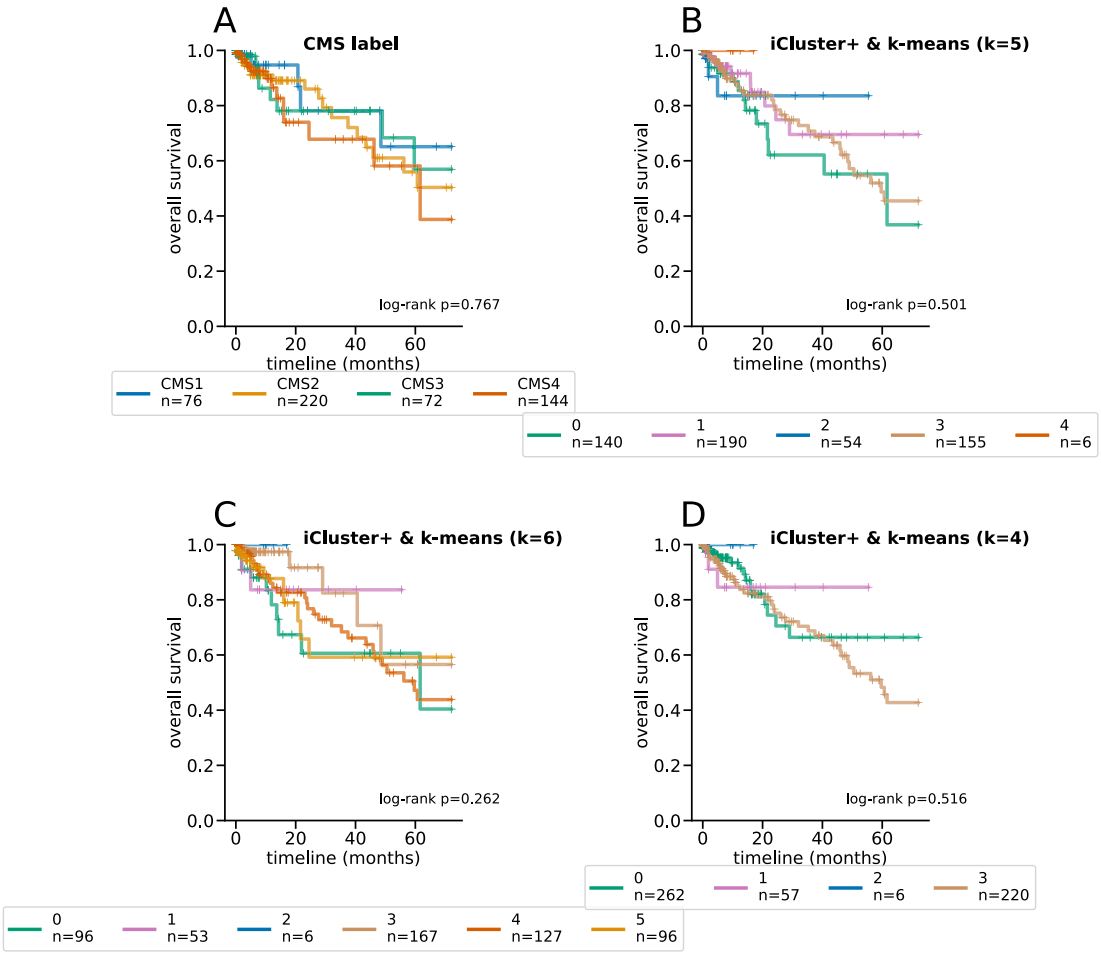


Figure B.5: Kaplan Meier curves and log-rank tests for differential survival statistics for the CMS subtypes, as well as iCluster+ clusters using k-means with different K's. The reported P values are from a multivariate log-rank test, under the null hypothesis that all groups have the same survival function. This figure is reproduced from Ronen et al. (2018).

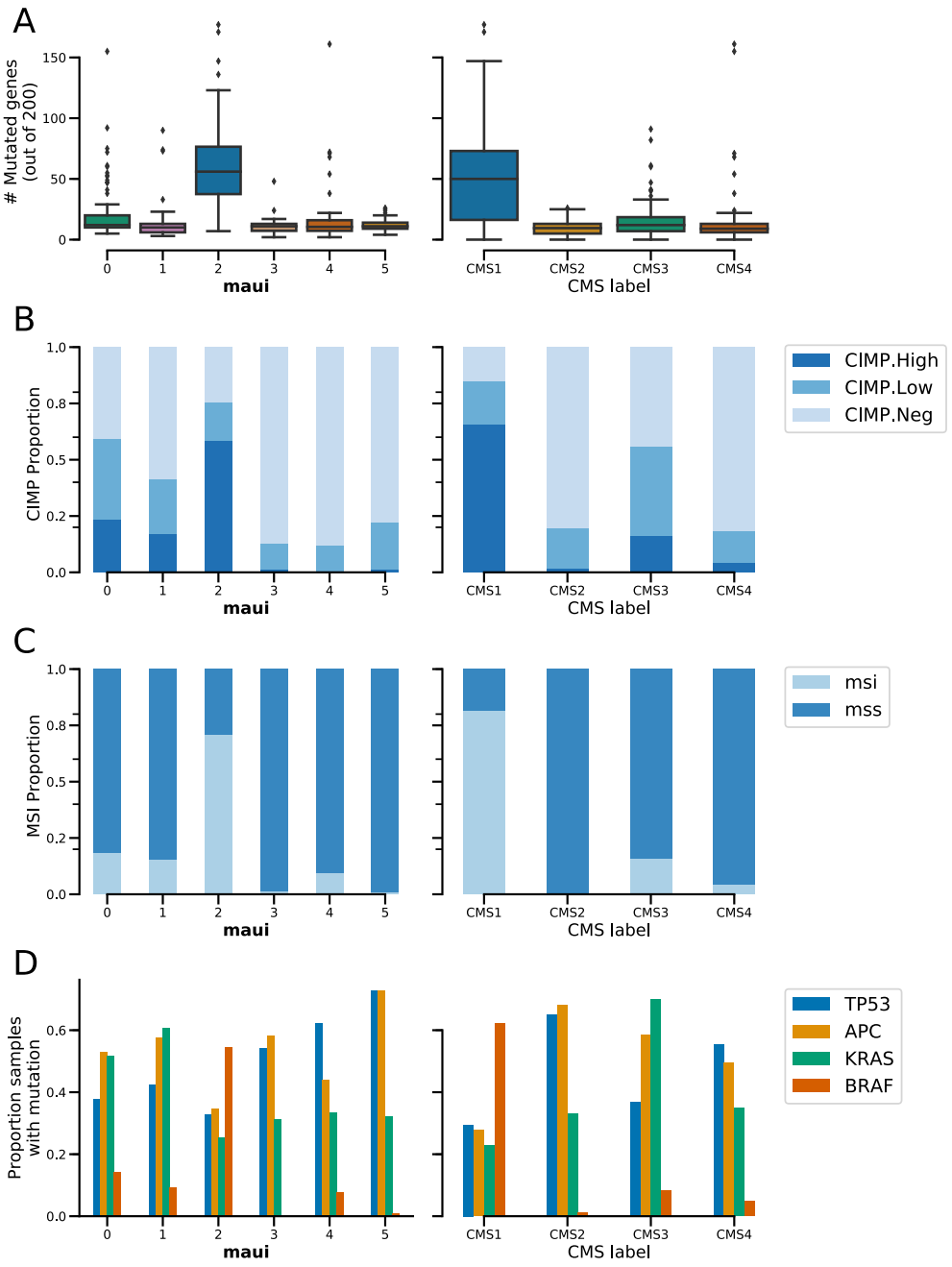


Figure B.6: Molecular markers and their distribution in CMS subtypes (left column) and maui clusters (right column). **A)** Mutational load, **B)** CIMP phenotype, **C)** Microsatellite instability, and **D)** the prevalence of mutations in a key set of colorectal cancer genes. This figure is reproduced from Ronen et al. (2018).

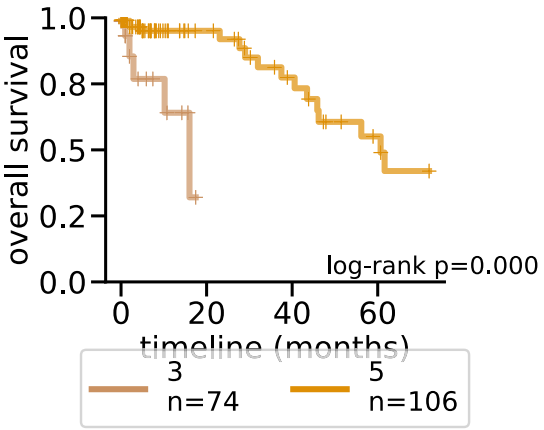


Figure B.7: Kaplan-Meier curves for maui clusters 3 and 5. Cluster 3 appears to be more aggressive tumors with a worse prognosis ($P < 0.001$). This figure is reproduced from Ronen et al. (2018).

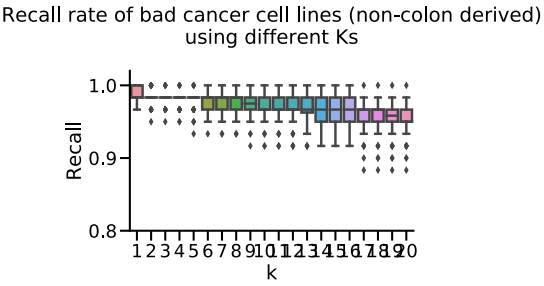


Figure B.8: We repeated the exercise of Figure 3.6E, that is, adding non-colon cell lines to the mix, and calculating the proportion of each cell line's K nearest neighbors, that are also cell lines (as opposed to tumors). Setting the threshold at 0.95, the method correctly identifies most non-colon cell lines as less likely to be appropriate models for colorectal tumors. The recall rate is $\frac{\text{\# models predicted to be less fit among non-colon cell lines}}{\text{\# non-colon cell lines}}$, and is largely insensitive to the choice of K. This figure is reproduced from Ronen et al. (2018).

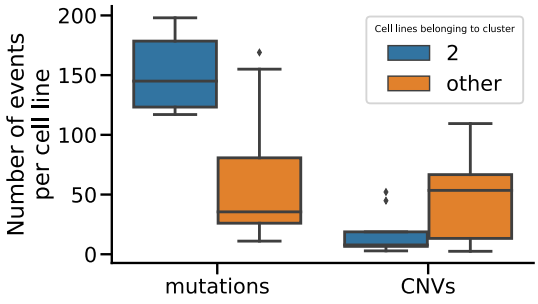


Figure B.9: The CMS1 subtype, which is captured by maui cluster 2, consists of hyper-mutated tumors with low chromosomal instability, resulting in tumors with a large number of mutations, but low number of copy number events (Figure B.6). The cell lines that we matched with cluster 2 also show the same characteristics. This figure is reproduced from Ronen et al. (2018).

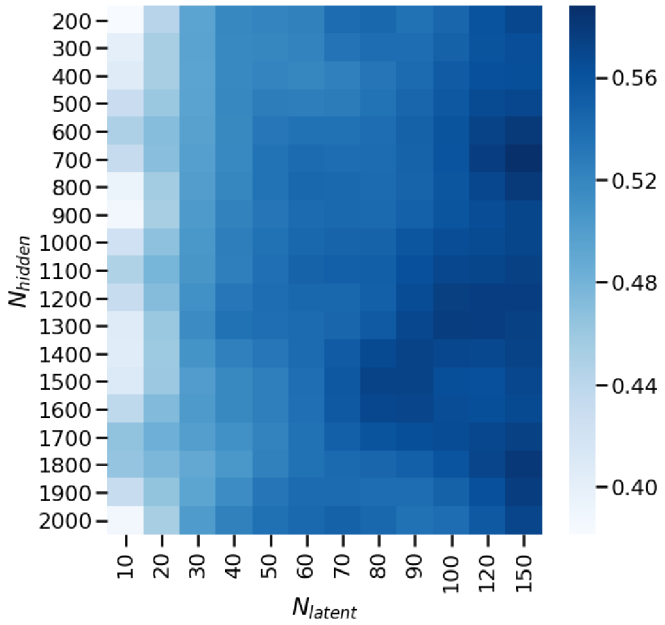


Figure B.10: The composite benchmark score in the space defined by N_{hidden}, the number of hidden units, and N_{latent}, the number of latent factors in a model. The optimal parameters are $N_{hidden} = 1500$ and $N_{latent}. = 80$ This figure is reproduced from Ronen et al. (2018).

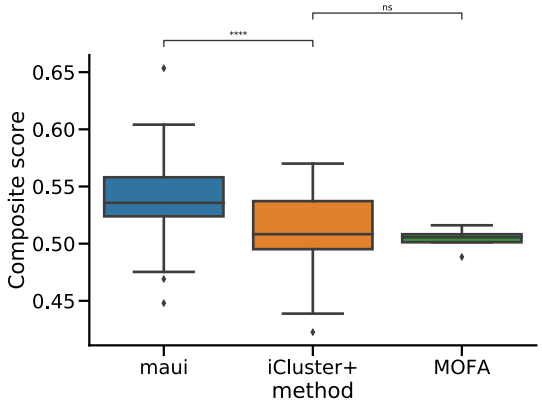


Figure B.11: We ran maui, iCluster+, and MOFA with a set of different parameters, performing a grid-search to find the best configuration. We computed a composite benchmark score (see Model selection and Figure B.10). This box plot shows the different results achieved by the different methods, demonstrating that maui tends to outperform iCluster+ and MOFA for a wide range of parameters. This figure is reproduced from Ronen et al. (2018).

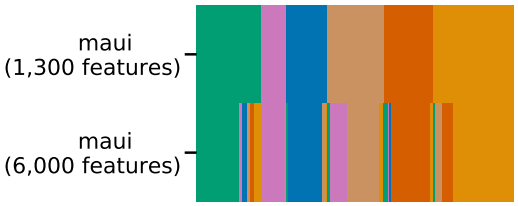


Figure B.12: Correspondence of maui clusters when training using 1,300 genes and 6,000 genes. Each column is a sample, and they are colored by their cluster assignment. Clusters are mostly the same when using more input features, with some refinements taking place. This figure is reproduced from Ronen et al. (2018).

References

10x Genomics (2017, 1). Single-cell rna-seq of 1.3 million brain cells from e18 mice. GEO accession GSE93421.

Ahmed, D., P. W. Eide, I. A. Eilertsen, S. A. Danielsen, M. Ekn?s, M. Hektoen, G. E. Lind, and R. A. Lothe (2013, Sep). Epigenetic and genetic features of 24 colon cancer cell lines. *Oncogenesis 2*, e71.

Akalin, A., V. Franke, B. Uyar, and J. Ronen (2019). *Computational Genomics with R*. Berlin, Germany.

Althuis, M. D., J. H. Fergenbaum, M. Garcia-Closas, L. A. Brinton, M. P. Madigan, and M. E. Sherman (2004). Etiology of hormone receptor–defined breast cancer: a systematic review of the literature. *Cancer Epidemiology and Prevention Biomarkers 13*(10), 1558–1568.

American Cancer Society (2019). Cancer facts and figures 2019. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2019/cancer-facts-and-figures-2019.pdf. Accessed: 2019-5-29.

Argelaguet, R., B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buettner, W. Huber, and O. Stegle (2018, jun). Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology 14*(6), e8124.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock (2000, May). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet. 25*(1), 25–29.

Bacher, R. and C. Kendziorski (2016, April). Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biology 17*(1).

Bahdanau, D., K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barski, A., S. Cuddapah, K. Cui, T.-Y. Roh, D. E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao (2007). High-resolution profiling of histone methylations in the human genome. *Cell 129*(4), 823–837.

Batzoglou, S., D. B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J. P. Mesirov, and E. S. Lander (2002). Arachne: a whole-genome shotgun assembler. *Genome research 12*(1), 177–189.

Ben-David, U., B. Siranosian, G. Ha, H. Tang, Y. Oren, K. Hinohara, C. A. Strathdee, J. Dempster, N. J. Lyons, R. Burns, A. Nag, G. Kugener, B. Cimini, P. Tsvetkov, Y. E. Maruvka, R. O'Rourke, A. Garrity, A. A. Tubelli, P. Bandopadhayay, A. Tsherniak, F. Vazquez, B. Wong, C. Birger, M. Ghandi, A. R. Thorner, J. A. Bittker, M. Meyerson, G. Getz, R. Beroukhim, and T. R. Golub (2018, aug). Genetic and transcriptional evolution alters cancer cell line drug response. *Nature 560*(7718), 325–330.

Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pp. 153–160.

Berg, K. C. G., P. W. Eide, I. A. Eilertsen, B. Johannessen, J. Bruun, S. A. Danielsen, M. Bj?rnslett, L. A. Meza-Zepeda, M. Ekn?s, G. E. Lind, O. Myklebost, R. I. Skotheim, A. Sveen, and R. A. Lothe (2017, 07). Multi-omics of 34 colorectal cancer cell lines - a resource for biomedical studies. *Mol. Cancer 16*(1), 116.

Beroukhim, R., G. Getz, L. Nghiemphu, J. Barretina, T. Hsueh, D. Linhart, I. Vivanco, J. C. Lee, J. H. Huang, S. Alexander, et al. (2007). Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. *Proceedings of the National Academy of Sciences 104*(50), 20007–20012.

Bhardwaj, N. and H. Lu (2005, jun). Correlation between gene expression profiles and protein-protein interactions within and across genomes. *Bioinformatics 21*(11), 2730–2738.

Blasius, J. (2006). *Multiple correspondence analysis and related methods*. Boca Raton: Chapman & Hall/CRC.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2016). Variational inference: A review for statisticians.

Breslow, N. E. (1975). Analysis of survival data under the proportional hazards model. *International Statistical Review / Revue Internationale de Statistique 43*(1), 45–57.

Buenrostro, J. D., P. G. Giresi, L. C. Zaba, H. Y. Chang, and W. J. Greenleaf (2013). Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods 10*(12), 1213.

Buhai, R.-D., A. Risteski, Y. Halpern, and D. Sontag (2019). Benefits of overparameterization in single-layer latent variable generative models.

Cadena, C., A. Dick, and I. D. Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation.

Cancer Genome Atlas Research Network et al. (2008). Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature 455*(7216), 1061.

Chen, C., A. Seff, A. Kornhauser, and J. Xiao (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2722–2730.

Chen, E. Y., C. M. Tan, Y. Kou, Q. Duan, Z. Wang, G. V. Meirelles, N. R. Clark, and A. Ma'ayan (2013, Apr). Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics 14*, 128.

Cheng, H., X. Yang, H. Si, A. D. Saleh, W. Xiao, J. Coupar, S. M. Gollin, R. L. Ferris, N. Issaeva, W. G. Yarbrough, M. E. Prince, T. E. Carey, C. V. Waes, and Z. Chen (2018, oct). Genomic and transcriptomic characterization links cell lines with aggressive head and neck cancers. *Cell Reports 25*(5), 1332–1345.e5.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Ciriello, G., M. L. Gatza, A. H. Beck, M. D. Wilkerson, S. K. Rhie, A. Pastore, H. Zhang, M. McLellan, C. Yau, C. Kandoth, et al. (2015). Comprehensive molecular portraits of invasive lobular breast cancer. *Cell 163*(2), 506–519.

Colaprico, A., T. C. Silva, C. Olsen, L. Garofano, C. Cava, D. Garolini, T. S. Sabedot, T. M. Malta, S. M. Pagnotta, I. Castiglioni, M. Ceccarelli, G. Bontempi, and H. Noushmehr (2016, 05). TCGAbiolinks: an R/Bioconductor package for integrative analysis of TCGA data. *Nucleic Acids Res. 44*(8), e71.

Consortium, T. G. O. (2017, Jan). Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res. 45*(D1), D331–D338.

de Tayrac, M., S. Le, M. Aubry, J. Mosser, and F. Husson (2009, Jan). Simultaneous analysis of distinct Omics data sets with integration of biological knowledge: Multiple Factor Analysis approach. *BMC Genomics 10*, 32.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*(1), 1–22.

Deng, Q., D. Ramsköld, B. Reinius, and R. Sandberg (2014, jan). Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science 343*(6167), 193–196.

DeRisi, J., L. Penland, M. Bittner, P. Meltzer, M. Ray, Y. Chen, Y. Su, and J. Trent (1996). Use of a cdna microarray to analyse gene expression. *Nat. genet 14*, 457–460.

Dey, S. S., L. Kester, B. Spanjaard, M. Bienko, and A. Van Oudenaarden (2015). Integrated genome and transcriptome sequencing of the same cell. *Nature biotechnology 33*(3), 285.

Dørum, G., L. Snipen, M. Solheim, and S. Saebo (2011, Aug). Smoothing gene expression data with network information improves consistency of regulated genes. *Stat Appl Genet Mol Biol 10*(1).

Eberwine, J., H. Yeh, K. Miyashiro, Y. Cao, S. Nair, R. Finnell, M. Zettel, and P. Coleman (1992). Analysis of gene expression in single live neurons. *Proceedings of the National Academy of Sciences 89*(7), 3010–3014.

Edgar, R., M. Domrachev, and A. E. Lash (2002, Jan). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res. 30*(1), 207–210.

ENCODE Project Consortium et al. (2004). The encode (encyclopedia of dna elements) project. *Science 306*(5696), 636–640.

ENCODE Project Consortium et al. (2007). Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature 447*(7146), 799.

Ernst, J. and M. Kellis (2012). Chromhmm: automating chromatin-state discovery and characterization. *Nature methods 9*(3), 215.

Fabregat, A., S. Jupe, L. Matthews, K. Sidiropoulos, M. Gillespie, P. Garapati, R. Haw, B. Jassal, F. Korninger, B. May, M. Milacic, C. D. Roca, K. Rothfels, C. Sevilla, V. Shamovsky, S. Shorser, T. Varusai, G. Viteri, J. Weiser, G. Wu, L. Stein, H. Hermjakob, and P. D'Eustachio (2018, Jan). The Reactome Pathway Knowledgebase. *Nucleic Acids Res. 46*(D1), D649–D655.

Fearon, E. R. (2011). Molecular genetics of colorectal cancer. *Annu Rev Pathol 6*, 479–507.

Felleman, D. J. and D. E. Van (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991) 1*(1), 1–47.

Fraser, H. B., A. E. Hirsh, D. P. Wall, and M. B. Eisen (2004, jun). Coevolution of gene expression among interacting proteins. *Proc Natl Acad Sci U S A 101*(24), 9033–9038.

Frommer, M., L. E. McDonald, D. S. Millar, C. M. Collis, F. Watt, G. W. Grigg, P. L. Molloy, and C. L. Paul (1992, March). A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands. *Proceedings of the National Academy of Sciences 89*(5), 1827–1831.

Gligorijević, V., M. Barot, and R. Bonneau (2018, jun). deepNF: deep network fusion for protein function prediction. *Bioinformatics*.

Gora, K. G., C. G. Tsokos, Y. E. Chen, B. S. Srinivasan, B. S. Perchuk, and M. T. Laub (2010). A cell-type-specific protein-protein interaction modulates transcriptional activity of a master regulator in caulobacter crescentus. *Molecular cell 39*(3), 455–467.

Grabherr, M. G., B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, et al. (2011). Full-length transcriptome assembly from rna-seq data without a reference genome. *Nature biotechnology 29*(7), 644.

Guinney, J., R. Dienstmann, X. Wang, A. de Reynies, A. Schlicker, C. Soneson, L. Marisa, P. Roepman, G. Nyamundanda, P. Angelino, B. M. Bot, J. S. Morris, I. M. Simon, S. Gerster, E. Fessler, F. De Sousa E Melo, E. Missiaglia, H. Ramay, D. Barras, K. Homicsko, D. Maru, G. C. Manyam, B. Broom, V. Boige, B. Perez-Villamil, T. Laderas, R. Salazar, J. W. Gray, D. Hanahan, J. Tabernero, R. Bernards, S. H. Friend, P. Laurent-Puig, J. P. Medema, A. Sadanandam, L. Wessels, M. Delorenzi, S. Kopetz, L. Vermeulen, and S. Tejpar (2015, Nov). The consensus molecular subtypes of colorectal cancer. *Nat. Med. 21*(11), 1350–1356. [PubMed Central:PMC4636487] [DOI:10.1038/nm.3967] [PubMed:26457759].

Harrell, F. E., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati (1982, May). Evaluating the yield of medical tests. *JAMA 247*(18), 2543–2546.

Harrell, F. E., K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati (1984). Regression modelling strategies for improved prognostic prediction. *Stat Med 3*(2), 143–152.

Harrell, F. E., K. L. Lee, and D. B. Mark (1996, Feb). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Stat Med 15*(4), 361–387.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Hausser, J. and K. Strimmer (2014). *entropy: Estimation of Entropy, Mutual Information and Related Quantities*. R package version 1.2.1.

Higgins, I., L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, Volume 3.

Hoadley, K. A., C. Yau, T. Hinoue, D. M. Wolf, A. J. Lazar, E. Drill, R. Shen, A. M. Taylor, A. D. Cherniack, V. Thorsson, et al. (2018). Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell 173*(2), 291–304.

Hoadley, K. A., C. Yau, D. M. Wolf, A. D. Cherniack, D. Tamborero, S. Ng, M. D. M. Leiserson, B. Niu, M. D. McLellan, V. Uzunangelov, J. Zhang, C. Kandoth, R. Akbani, H. Shen, L. Omberg, A. Chu, A. A. Margolin, L. J. Van't Veer, N. Lopez-Bigas, P. W. Laird, B. J. Raphael, L. Ding, A. G. Robertson, L. A. Byers, G. B. Mills, J. N. Weinstein, C. Van Waes, Z. Chen, E. A. Collisson, C. C. Benz, C. M. Perou, and J. M. Stuart (2014, Aug). Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell 158*(4), 929–944.

Hofree, M., J. P. Shen, H. Carter, A. Gross, and T. Ideker (2013, Nov). Network-based stratification of tumor mutations. *Nat. Methods 10*(11), 1108–1115.

Hubel, D. H. and T. N. Wiesel (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology 160*(1), 106–154.

International Human Genome Sequencing Consortium et al. (2004). Finishing the euchromatic sequence of the human genome. *Nature 431*(7011), 931.

Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Islam, S., U. Kjällquist, A. Moliner, P. Zajac, J.-B. Fan, P. Lönnerberg, and S. Linnarsson (2011). Characterization of the single-cell transcriptional landscape by highly multiplex rna-seq. *Genome research 21*(7), 1160–1167.

Jobling, P., J. Pundavela, S. M. Oliveira, S. Roselli, M. M. Walker, and H. Hondermarck (2015, May). Nerve-Cancer Cell Cross-talk: A Novel Promoter of Tumor Progression. *Cancer Res. 75*(9), 1777–1781.

Johnson, D. S., A. Mortazavi, R. M. Myers, and B. Wold (2007). Genome-wide mapping of in vivo protein-dna interactions. *Science 316*(5830), 1497–1502.

Kalchbrenner, N. and P. Blunsom (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709.

Kanehisa, M. (2000, jan). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research 28*(1), 27–30.

Kanehisa, M., M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima (2016, nov). KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research 45*(D1), D353–D361.

Kanehisa, M., M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima (2017, Jan). KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res. 45*(D1), D353–D361.

Kanehisa, M., Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe (2015, oct). KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research 44*(D1), D457–D462.

Kharchenko, P. V., L. Silberstein, and D. T. Scadden (2014a). Bayesian approach to single-cell differential expression analysis. *Nature methods 11*(7), 740.

Kharchenko, P. V., L. Silberstein, and D. T. Scadden (2014b, jul). Bayesian approach to single-cell differential expression analysis. *Nat Methods 11*(7), 740–742.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization.

Kingma, D. P. and M. Welling (2013). Auto-encoding variational bayes.

Kiros, R., R. Salakhutdinov, and R. S. Zemel (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.

Kitadai, Y., T. Sasaki, T. Kuwai, T. Nakamura, C. D. Bucana, S. R. Hamilton, and I. J. Fidler (2006, Dec). Expression of activated platelet-derived growth factor receptor in stromal cells of human colon carcinomas is associated with metastatic potential. *Int. J. Cancer 119*(11), 2567–2574.

Klein, A. M., L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell 161*(5), 1187–1201.

Kuipers, E. J., W. M. Grady, D. Lieberman, T. Seufferlein, J. J. Sung, P. G. Boelens, C. J. van de Velde, and T. Watanabe (2015, 11). Colorectal cancer. *Nat Rev Dis Primers 1*, 15065.

Kuleshov, M. V., M. R. Jones, A. D. Rouillard, N. F. Fernandez, Q. Duan, Z. Wang, S. Koplev, S. L. Jenkins, K. M. Jagodnik, A. Lachmann, M. G. McDermott, C. D. Monteiro, G. W. Gundersen, and A. Ma'ayan (2016, 07). Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res. 44*(W1), W90–97.

Lao, V. V. and W. M. Grady (2011, oct). Epigenetics and colorectal cancer. *Nature Reviews Gastroenterology & Hepatology 8*(12), 686–700.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *nature 521*(7553), 436.

Lee, I., U. M. Blom, P. I. Wang, J. E. Shim, and E. M. Marcotte (2011a, jul). Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Res 21*(7), 1109–1121.

Lee, I., U. M. Blom, P. I. Wang, J. E. Shim, and E. M. Marcotte (2011b, Jul). Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Res. 21*(7), 1109–1121.

Li, W. V. and J. J. Li (2017). scimpute: Accurate and robust imputation for single cell rna-seq data. *bioRxiv*.

Liebig, C., G. Ayala, J. Wilks, G. Verstovsek, H. Liu, N. Agarwal, D. H. Berger, and D. Albo (2009, Nov). Perineural invasion is an independent predictor of outcome in colorectal cancer. *J. Clin. Oncol. 27*(31), 5131–5137.

Lin, P., M. Troup, and J. W. K. Ho (2017, mar). Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biol 18*(1), 59.

Louhimo, R. and S. Hautaniemi (2011). Cnamet: an r package for integrating copy number, methylation and expression data. *Bioinformatics 27*(6), 887–888.

Macosko, E. Z., A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, et al. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell 161*(5), 1202–1214.

Margulies, M., M. Egholm, W. E. Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y.-J. Chen, Z. Chen, et al. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature 437*(7057), 376.

Maxam, A. M. and W. Gilbert (1977). A new method for sequencing dna. *Proceedings of the National Academy of Sciences 74*(2), 560–564.

McCarthy, D. J., K. R. Campbell, A. T. L. Lun, and Q. F. Wills (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics 33*(8), 1179–1186.

McFarland, C. D., J. A. Yaglom, J. W. Wojtkowiak, J. G. Scott, D. L. Morse, M. Y. Sherman, and L. A. Mirny (2017, may). The damaging effect of passenger mutations on cancer progression. *Cancer Research*.

McInnes, L. and J. Healy (2018). Umap: Uniform manifold approximation and projection for dimension reduction.

Medico, E., M. Russo, G. Picco, C. Cancelliere, E. Valtorta, G. Corti, M. Buscarino, C. Isella, S. Lamba, B. Martinoglio, S. Veronese, S. Siena, A. Sartore-Bianchi, M. Beccuti, M. Mottolese, M. Linnebacher, F. Cordero, F. Di Nicolantonio, and A. Bardelli (2015, Apr). The molecular landscape of colorectal cancer cell lines unveils clinically actionable kinase targets. *Nat Commun 6*, 7002.

Mikkelsen, T. S., M. Ku, D. B. Jaffe, B. Issac, E. Lieberman, G. Giannoukos, P. Alvarez, W. Brockman, T.-K. Kim, R. P. Koche, et al. (2007). Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature 448*(7153), 553.

Million Women Study Collaborators et al. (2003). Breast cancer and hormone-replacement therapy in the million women study. *The Lancet 362*(9382), 419–427.

Mo, Q. and R. Shen (2013). iclusterplus: integrative clustering of multiple genomic data sets.

Mo, Q., S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, C. Sander, R. S. Powers, M. Ladanyi, and R. Shen (2013, Mar). Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl. Acad. Sci. U.S.A. 110*(11), 4245–4250.

Morin, R. D., M. Bainbridge, A. Fejes, M. Hirst, M. Krzywinski, T. J. Pugh, H. McDonald, R. Varhol, S. J. Jones, and M. A. Marra (2008). Profiling the hela s3 transcriptome using randomly primed cdna and massively parallel short-read sequencing. *Biotechniques 45*(1), 81–94.

Mortazavi, A., B. A. Williams, K. McCue, L. Schaeffer, and B. Wold (2008). Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods 5*(7), 621.

Müller, M. F., A. E. K. Ibrahim, and M. J. Arends (2016, jun). Molecular pathological classification of colorectal cancer. *Virchows Archiv 469*(2), 125–134.

Nestorowa, S., F. K. Hamey, B. Pijuan Sala, E. Diamanti, M. Shepherd, E. Laurenti, N. K. Wilson, D. G. Kent, and B. Göttgens (2016, aug). A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood 128*(8), e20–31.

Oshlack, A., M. D. Robinson, and M. D. Young (2010). From rna-seq reads to differential expression results. *Genome biology 11*(12), 220.

Parsons, D. W., T. L. Wang, Y. Samuels, A. Bardelli, J. M. Cummins, L. DeLong, N. Silliman, J. Ptak, S. Szabo, J. K. Willson, S. Markowitz, K. W. Kinzler, B. Vogelstein, C. Lengauer, and V. E. Velculescu (2005, Aug). Colorectal cancer: mutations in a signalling pathway. *Nature 436*(7052), 792.

Patel, A. P., I. Tirosh, J. J. Trombetta, A. K. Shalek, S. M. Gillespie, H. Wakimoto, D. P. Cahill, B. V. Nahed, W. T. Curry, R. L. Martuza, D. N. Louis, O. Rozenblatt-Rosen, M. L. Suvà, A. Regev, and B. E. Bernstein (2014, jun). Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science 344*(6190), 1396–1401.

Pencina, M. J. and R. B. D'Agostino (2004, jun). OverallC as a measure of discrimination in survival analysis: model specific population value and confidence interval estimation. *Statistics in Medicine 23*(13), 2109–2123.

Petryszak, R., M. Keays, Y. A. Tang, N. A. Fonseca, E. Barrera, T. Burdett, A. Füllgrabe, A. M.-P. Fuentes, S. Jupp, S. Koskinen, O. Mannion, L. Huerta, K. Megy, C. Snow, E. Williams, M. Barzine, E. Hastings, H. Weisser, J. Wright, P. Jaiswal, W. Huber, J. Choudhary, H. E. Parkinson, and A. Brazma (2016). Expression atlas update—an integrated database of gene and protein expression in humans, animals and plants. *Nucleic Acids Research 44*(D1), D746–D752.

Pierson, E. and C. Yau (2015, nov). Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol 16*, 241.

Purdom, E. and D. Risso (2017). *clusterExperiment: Compare Clusterings for Single-Cell Sequencing*. R package version 1.2.0.

Regev, A., S. Teichmann, O. Rozenblatt-Rosen, M. Stubbington, K. Ardlie, I. Amit, P. Arlotta, G. Bader, C. Benoist, M. Biton, B. Bodenmiller, B. Bruneau, P. Campbell, M. Carmichael, P. Carninci, L. Castelo-Soccio, M. Clatworthy, H. Clevers, C. Conrad, R. Eils, J. Freeman, L. Fugger, B. Goettgens, D. Graham, A. Greka, N. Hacohen, M. Haniffa, I. Helbig, R. Heuckeroth, S. Kathiresan, S. Kim, A. Klein, B. Knoppers, A. Kriegstein, E. Lander, J. Lee, E. Lein, S. Linnarsson, E. Macosko, S. MacParland, R. Majovski, P. Majumder, J. Marioni, I. McGilvray, M. Merad, M. Mhlanga, S. Naik, M. Nawijn, G. Nolan, B. Paten, D. Pe'er, A. Philippakis, C. Ponting, S. Quake, J. Rajagopal, N. Rajewsky, W. Reik, J. Rood, K. Saeb-Parsy, H. Schiller, S. Scott, A. Shalek, E. Shapiro, J. Shin, K. Skeldon, M. Stratton, J. Streicher, H. Stunnenberg, K. Tan, D. Taylor, A. Thorogood, L. Vallier, A. van Oudenaarden, F. Watt, W. Weicher, J. Weissman, A. Wells, B. Wold, R. Xavier, X. Zhuang, and H. C. A. O. Committee (2018). The human cell atlas white paper.

Robertson, G., M. Hirst, M. Bainbridge, M. Bilenky, Y. Zhao, T. Zeng, G. Euskirchen, B. Bernier, R. Varhol, A. Delaney, et al. (2007). Genome-wide profiles of stat1 dna association using chromatin immunoprecipitation and massively parallel sequencing. *Nature methods 4*(8), 651.

Robinson, M. D., D. J. McCarthy, and G. K. Smyth (2010, jan). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics 26*(1), 139–140.

Ronaghi, M., S. Karamohamed, B. Pettersson, M. Uhlén, and P. Nyrén (1996). Real-time dna sequencing using detection of pyrophosphate release. *Analytical biochemistry 242*(1), 84–89.

Ronen, J. and A. Akalin (2018a). netSmooth: Network-smoothing based imputation for single cell RNA-seq. *F1000Res 7*, 8.

Ronen, J. and A. Akalin (2018b). *netSmooth: Network smoothing for scRNAseq*. R package version 1.0.

Ronen, J., S. Hayat, and A. Akalin (2018). Evaluation of colorectal cancer subtypes and cell lines using deep learning. *bioRxiv*.

Sanger, F., S. Nicklen, and A. R. Coulson (1977, December). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences 74*(12), 5463–5467.

Shankland, S. Google translate now serves 200 million people daily - cnet.

Shen, R., Q. Mo, N. Schultz, V. E. Seshan, A. B. Olshen, J. Huse, M. Ladanyi, and C. Sander (2012). Integrative subtype discovery in glioblastoma using icluster. *PloS one 7*(4), e35236.

Solomon, M. J., P. L. Larsen, and A. Varshavsky (1988). Mapping proteindna interactions in vivo with formaldehyde: evidence that histone h4 is retained on a highly transcribed gene. *Cell 53*(6), 937–947.

Soneson, C. and M. D. Robinson (2017). Bias, robustness and scalability in differential expression analysis of single-cell rna-seq data. *bioRxiv*.

Sood, P., A. Krek, M. Zavolan, G. Macino, and N. Rajewsky (2006). Cell-type-specific signatures of micrornas on target mrna expression. *Proceedings of the National Academy of Sciences 103*(8), 2746–2751.

Staden, R. (1979). A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research 6*(7), 2601–2610.

Steller, E. J., D. A. Raats, J. Koster, B. Rutten, K. M. Govaert, B. L. Emmink, N. Snoeren, S. R. van Hooff, F. C. Holstege, C. Maas, I. H. Borel Rinkes, and O. Kranenburg (2013, Feb). PDGFRB promotes liver metastasis formation of mesenchymal-like colorectal tumor cells. *Neoplasia 15*(2), 204–217.

Sun, C., S. Hobor, A. Bertotti, D. Zecchin, S. Huang, F. Galimi, F. Cottino, A. Prahallad, W. Grernrum, A. Tzani, A. Schlicker, L. F. Wessels, E. F. Smit, E. Thunnissen, P. Halonen, C. Lieftink, R. L. Beijersbergen, F. Di Nicolantonio, A. Bardelli, L. Trusolino, and

R. Bernards (2014, Apr). Intrinsic resistance to MEK inhibition in KRAS mutant lung and colon cancer through transcriptional induction of ERBB3. *Cell Rep 7*(1), 86–93.

Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112.

Svensson, V. (2019). Droplet scrna-seq is not zero-inflated. *bioRxiv*.

Szklarczyk, D., J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, L. J. Jensen, and C. von Mering (2016, oct). The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research 45*(D1), D362–D368.

Szklarczyk, D., J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, L. J. Jensen, and C. von Mering (2017, jan). The string database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res 45*(D1), D362–D368.

Sønderby, C. K., T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther (2016). Ladder variational autoencoders.

Tang, F., C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, et al. (2009). mrna-seq whole-transcriptome analysis of a single cell. *Nature methods 6*(5), 377.

The Cancer Genome Atlas Network (2012). Comprehensive molecular characterization of human colon and rectal cancer. *Nature 487*(7407), 330.

Tietjen, I., J. M. Rihel, Y. Cao, G. Koentges, L. Zakhary, and C. Dulac (2003). Single-cell transcriptional analysis of neuronal progenitors. *Neuron 38*(2), 161–175.

Tini, G., L. Marchetti, C. Priami, and M.-P. Scott-Boyer (2017, dec). Multi-omics integration—a comparison of unsupervised clustering methodologies. *Briefings in Bioinformatics*.

Toyota, M., N. Ahuja, M. Ohe-Toyota, J. G. Herman, S. B. Baylin, and J.-P. J. Issa (1999, jul). CpG island methylator phenotype in colorectal cancer. *Proceedings of the National Academy of Sciences 96*(15), 8681–8686.

Trapnell, C., D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology 32*(4), 381.

Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman (2001). Missing value estimation methods for dna microarrays. *Bioinformatics 17*(6), 520–525.

Trunk, G. V. (1979, jul). A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1*(3), 306–307.

Tyagi, S. and F. R. Kramer (1996). Molecular beacons: probes that fluoresce upon hybridization. *Nature biotechnology 14*(3), 303.

van der Maaten, L. and G. Hinton (2008). Visualizing high-dimensional data using t-sne.

van Dijk, D., J. Nainys, R. Sharma, P. Kathail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er (2017). Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. *bioRxiv*.

Vandin, F., E. Upfal, and B. J. Raphael (2011, mar). Algorithms for detecting significantly mutated pathways in cancer. *J Comput Biol 18*(3), 507–522.

Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol (2010, December). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res. 11*, 3371–3408.

Vinh, N. X., J. Epps, and J. Bailey (2010, December). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res. 11*, 2837–2854.

Vinyals, O., A. Toshev, S. Bengio, and D. Erhan (2016). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence 39*(4), 652–663.

Vlachogiannis, G., S. Hedayat, A. Vatsiou, Y. Jamin, J. Fernández-Mateos, K. Khan, A. Lampis, K. Eason, I. Huntingford, R. Burke, M. Rata, D.-M. Koh, N. Tunariu, D. Collins, S. Hulkki-Wilson, C. Ragulan, I. Spiteri, S. Y. Moorcraft, I. Chau, S. Rao, D. Watkins, N. Fotiadis, M. Bali, M. Darvish-Damavandi, H. Lote, Z. Eltahir, E. C. Smyth, R. Begum, P. A. Clarke, J. C. Hahne, M. Dowsett, J. de Bono, P. Workman, A. Sadanandam, M. Fassan, O. J. Sansom, S. Eccles, N. Starling, C. Braconi, A. Sottoriva, S. P. Robinson, D. Cunningham, and N. Valeri (2018, feb). Patient-derived organoids model treatment response of metastatic gastrointestinal cancers. *Science 359*(6378), 920–926.

Wagner, A., A. Regev, and N. Yosef (2016, nov). Revealing the vectors of cellular identity with single-cell genomics. *Nat Biotechnol 34*(11), 1145–1160.

Way, G. P. and C. S. Greene (2017). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *bioRxiv*.

Winter, R., F. Montanari, F. Noé, and D.-A. Clevert (2019). Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci. 10*, 1692–1701.

Wreczycka, K., V. Franke, B. Uyar, R. Wurmus, S. Bulut, B. Tursun, and A. Akalin (2019, 05). HOT or not: examining the basis of high-occupancy target regions. *Nucleic Acids Research*.

Wreczycka, K., A. Gosdschan, D. Yusuf, B. Grüning, Y. Assenov, and A. Akalin (2017). Strategies for analyzing bisulfite sequencing data. *Journal of biotechnology 261*, 105–115.

Wu, A. R., N. F. Neff, T. Kalisky, P. Dalerba, B. Treutlein, M. E. Rothenberg, F. M. Mburu, G. L. Mantalas, S. Sim, M. F. Clarke, and S. R. Quake (2014, jan). Quantitative assessment of single-cell rna-sequencing methods. *Nat Methods 11*(1), 41–46.

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wurmus, R., B. Uyar, B. Osberg, V. Franke, A. Gosdschan, K. Wreczycka, J. Ronen, and A. Akalin (2018). Pigx: reproducible genomics analysis pipelines with gnu guix. *GigaScience 7*(12), giy123.

Yang, Z. and G. Michailidis (2015). A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics 32*(1), 1–8.

Zhao, S., W.-P. Fung-Leung, A. Bittner, K. Ngo, and X. Liu (2014). Comparison of rna-seq and microarray in transcriptome profiling of activated t cells. *PloS one 9*(1), e78644.